Utilizing Static Packet Filters to Enhance Network Security

GIAC Security Essentials
Certification (GSEC)
Practical Assignment
Version 1.4B

Option 1 - Research on Topics
in Information Security

Submitted by: Scott Foster
Date: 10/22/2004
Location: Sandy, UT

Paper Abstract: Many network installations today consist of a firewall to provide security between the increasing hostile environment of the Internet and the corporate network. This paper examines utilizing Access Control Lists to implement static packet filters at a network perimeter to enhance security in any sized network. An examination of NSA recommended filters will be performed, potential weaknesses discussed, enhancements conceived, and the effects of these filters upon the devices they are placed. Additional ideas will include enhancements to Cisco routers that are available to provide further security.

Table of Contents

List of Figures

Abstract/Summary

Many network installations today consist of a firewall to provide security between the increasing hostile environment of the Internet and the corporate network. This paper examines utilizing Access Control Lists to implement static packet filters at a network perimeter to enhance security in any sized network. An examination of NSA recommended filters will be performed, potential weaknesses discussed, enhancements conceived, and the effects of these filters upon the devices they are placed. Additional ideas will include enhancements to Cisco routers that are available to provide further security.

Introduction

Many network installations today do not include adequate security between the increasing hostile environment of the Internet and the shelter from this storm; the corporate network. There are many reasons that may contribute to this lax security approach. Some of these reasons might include: funding issues, device control, political issues, management, ignorance or outright incompetence. Whatever the reason, for any organization to have an Internet presence, a practice of defense-in-depth must be implemented.

An article, "Security in the Branch or Small Office," in the 4th quarter volume of the magazine Packet describes this issue. The author, Janet Kreiling, writes in regards to branch and small offices of any organization, "Standalone offices often lack the staff and budget to prepare for or respond effectively to a security breach. Branch offices, while theoretically having access to enterprise IT and security staff may, in reality, sit as poorly defended outposts that can provide unauthorized access to company operations and databases."

The same article quotes the director of engineering for access router security at Cisco, Adrian Amelse, concerning defense measures, "The access router can deliver comprehensive security technologies including a stateful firewall, intrusion detection, access control lists, virtual private networks, and others, as well as networking features such as rate-limiting that also boost security. The fact that these technologies are already in a system that is part of a packet's path into or out of your office improves packet security and minimizes latency for delay-sensitive applications. And they are easy to bring online." Adrian continues describing these security features in detail including access control lists (ACLs) which are available on routers that allow or deny network traffic based on their addresses and services. He specifically points out that an ACL, "is useful in combating something like the Slammer worm, which homed in specifically on port number 1434."

This paper examines utilizing Access Control Lists on Cisco routers to implement static packet filters with respect to incoming/outgoing, IP-based traffic in order to enhance security in any sized network. There are several advantages to using Cisco ACLs to implement an extra layer of security. The first of these advantages is that most Cisco routers provide the use of ACLs for packet filtering capability. Second, the overhead placed upon a Cisco router to implement this extra layer of security is minimal as will be examined later. Another bonus of utilizing Cisco ACLs is the speed at which a packet filter may operate in order to provide an extra layer of security. According to a paper entitled "Evolution of the Firewall Industry," by Cisco Systems "it [static packet filtering] does less processing than other technologies…it is the fastest firewall technology available." Since Cisco ACLs don't examine a packet's "state" by default, packets can be filtered extremely fast to minimize connection delays and latency. Finally, Cisco ACLs are extensible and can be implemented with stateful features that provide a finer level of granularity for filtering.

Using the “Router Configuration Security Guide,” distributed by the National Security Agency or “NSA” as a template, a detailed rule-by-rule examination of a NSA recommended ACL will be performed. Each line of the NSA based ACL will be discussed in depth with a break down of the rule(s) being implemented and examples to provide further insight. The weaknesses or issues associated with this base filter will be examined. Enhancements to mitigate these issues will also be proposed. In addition, data will be presented to identify the impact on a router’s resources after an ACL is implemented.

The NSA Based ACL

The National Security Agency (www.nsa.org) provides an in-depth document on securing routers through, among other technologies, the used of access control lists (ACLs). This document, "The NSA Router Security Guide," submits that the use of ACLs at the network perimeter provides an initial layer of defense into an organization's network. Many security specialists agree to the ability that an ACL has for perimeter security. According to the article "Safe at Any Speed?" in the July 2002 issue of Information Security magazine written by Ray Kaplan, "Properly configured, routers can be the perimeter's first line of security defense. A suitable router will allow granular control of network traffic. For instance, it's common to configure a router to protect against spoofing."

The NSA identifies what rules should be included in an ACL to deny traffic, in-or-outbound, that could potential compromise the integrity of an internal network. After it's discussion, the paper provides an example of an ACL to accomplish this. Upon examination, the ACL appears to provide a balance between administrative-ease and the practice of "least privilege." The result is a list that filters much of the "noise" that the Internet produces without becoming administratively burdensome.

Below is a picture diagramming the network configured to test the use of ACLs. An ACL based on the NSA example and customized to this environment was conceived and will be presented in the next section. While the examples presented in this paper are based on Cisco ACLs, the concepts discussed can be applied to many other manufacturers of routers, firewalls and layer 3 switches.

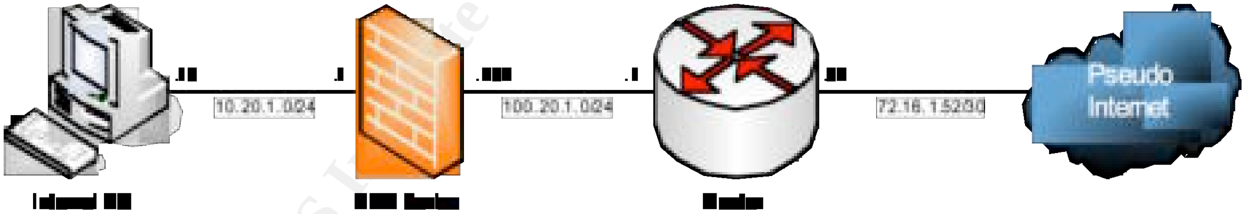


Figure 1

Network Description

- A Cisco router providing access to the Internet
  - External Interface IP: 72.16.1.52/30
  - Internal Interface IP: 100.20.1.1/24
- A firewall performing NAT translation
  - External Interface IP: 100.20.1.254/24
  - Internet Interface IP: 10.20.1.1/24
  - Externally addressable IP to Internal Address: 100.20.1.5 to 10.20.1.10
  - Policy to permit all traffic in and outbound
- A PC representing the internal network of an organization
  - Windows 2000 Server (No service packs installed)
  - Externally available services: HTTP, FTP, DNS, SMTP
  - Access to external services needed: HTTP, HTTPS, FTP, DNS, SMTP, NTP

The Proposed ACL

The ACL below might be considered by some as a “restrictive filter” as described in an article by The SCO Group. “You can construct a filter so that it only allows packets destined for specified services to pass. All other packets are dropped. This approach is best if you can easily specify which services you want to allow such as HTTP and DNS. You may inadvertently block certain services that people want to use but you can add these later if necessary.” This ACL attempts to implement this method of filtering as only packets destined to specific externally accessible services are permitted inbound.

```
access-list 100 remark This access-list filters ingress traffic from an
untrusted source
access-list 100 deny ip 100.20.1.0 0.0.0.255 any log
access-list 100 deny ip host 72.16.1.54 host 72.16.1.54 log
access-list 100 deny ip 127.0.0.0 0.255.255.255 any log
access-list 100 deny ip 0.0.0.0 0.255.255.255 any log
access-list 100 deny ip 10.0.0.0 0.255.255.255 any log
access-list 100 deny ip 172.16.0.0 0.15.255.255 any log
access-list 100 deny ip 192.168.0.0 0.0.255.255 any log
access-list 100 deny ip 169.254.0.0 0.0.255.255 any log
access-list 100 deny ip 192.0.2.0 0.0.0.255 any log
access-list 100 deny ip 224.0.0.0 15.255.255.255 any log
access-list 100 deny ip any host 100.20.1.255 log
access-list 100 deny ip any host 100.20.1.0 log
access-list 100 deny ip host 255.255.255.255 any log
access-list 100 permit tcp any 100.20.1.0 0.0.0.255 established
access-list 100 deny icmp any any echo log
access-list 100 deny icmp any any redirect log
access-list 100 deny icmp any any mask-request log
access-list 100 permit icmp any 100.20.1.0 0.0.0.255
access-list 100 deny tcp any any range 6000 6063 log
access-list 100 deny tcp any any eq 6667 log
access-list 100 deny tcp any any range 12345 12346 log
access-list 100 deny tcp any any eq 31337 log
access-list 100 deny udp any any eq 2049 log
access-list 100 deny udp any any eq 31337 log
access-list 100 permit tcp any gt 1023 100.20.1.0 0.0.0.255 eq ftp
access-list 100 permit tcp any gt 1023 100.20.1.0 0.0.0.255 eq www
access-list 100 permit tcp any gt 1023 100.20.1.0 0.0.0.255 eq smtp
access-list 100 permit tcp any gt 1023 100.20.1.0 0.0.0.255 eq 443
access-list 100 deny udp any any range 33400 34400 log
access-list 100 permit udp any eq domain 100.20.1.0 0.0.0.255 gt 1023
access-list 100 permit udp any eq domain 100.20.1.0 0.0.0.255 eq domain
access-list 100 permit udp any gt 1023 100.20.1.0 0.0.0.255 gt domain
access-list 100 permit tcp any gt 1023 10.20.1.0.0.0.0.255 eq domain
access-list 100 permit udp any eq ntp 100.20.1.0 0.0.0.255 eq ntp
access-list 100 permit udp any eq ntp 100.20.1.0 0.0.0.255 gt 1023
access-list 100 permit udp any gt 1023 100.20.1.0 0.0.0.255 eq ntp
access-list 100 permit tcp any gt 1023 100.20.1.0 0.0.0.255 range 1024 5000
access-list 100 deny tcp any range 0 65535 any range 0 65535 log
access-list 100 deny udp any range 0 65535 any range 0 65535 log
access-list 100 deny ip any any log
```

```
access-list 102 remark This access-list filters egress traffic to an
untrusted destination
access-list 102 deny ip host 100.20.1.0 host 100.20.1.0 log
access-list 102 permit icmp 100.20.1.0 0.0.0.255 any echo-reply
access-list 102 permit icmp 100.20.1.0 0.0.0.255 any parameter-problem
access-list 102 permit icmp 100.20.1.0 0.0.0.255 any packet-too-big
access-list 102 permit icmp 100.20.1.0 0.0.0.255 any source-quench
access-list 102 deny tcp any any range 1 19 log
access-list 102 deny tcp any any eq 43 log
access-list 102 deny tcp any any eq 93 log
access-list 102 deny tcp any any range 135 139 log
access-list 102 deny tcp any any eq 445 log
access-list 102 deny tcp any any range 512 518 log
access-list 102 deny tcp any any eq 540 log
access-list 102 permit tcp 100.20.1.0 0.0.0.255 gt 1023 any
access-list 102 permit tcp 100.20.1.0 0.0.0.255 eq www any gt 1023
established
access-list 102 permit tcp 100.20.1.0 0.0.0.255 eq smtp any gt 1023
established
access-list 102 permit tcp 100.20.1.0 0.0.0.255 eq 443 any gt 1023
established
access-list 102 permit tcp 100.20.1.0 0.0.0.255 eq ftp any gt 1023
established
access-list 102 permit tcp 100.20.1.0 0.0.0.255 range 1024 5000 ftp-data any
gt 1023 established
access-list 102 permit udp 100.20.1.0 0.0.0.255 gt 1023 any eq domain
access-list 102 permit udp 100.20.1.0 0.0.0.255 eq domain any gt 1023
access-list 102 permit udp 100.20.1.0 0.0.0.255 eq domain any eq domain
access-list 102 permit udp 100.20.1.0 0.0.0.255 gt 1023 any eq ntp
access-list 102 permit udp 100.20.1.0 0.0.0.255 eq ntp any eq ntp
access-list 102 permit udp 100.20.1.0 0.0.0.255 any range 33400 34400 log
access-list 102 deny tcp any range 0 65535 any range 0 65535 log
access-list 102 deny udp any range 0 65535 any range 0 65535 log
access-list 102 deny ip any any log
```

Analysis

This section will contain a rule-by-rule analysis of each access-list entry. This analysis will provide a better understanding of the rules that make up the ACL. Some entries may be combined together because of their relation to each other in the list.

```
access-list 100 remark This access-list filters ingress traffic from an
untrusted source
```

Depending on the use of a router, it could be configured with many ACLs for various functions. Adding a remark to the beginning of an access-list will assist a router administrator to identify what the access-list is used for. Additional remarks may be included within the body of the access-list to provide clarity if it is warranted.

```
access-list 100 deny ip 100.20.1.0 0.0.0.255 any log
```

Since the organization is hosting the 100.20.1.0/24 network, there should not be any traffic coming from the Internet identifying itself with a source address belonging to this network. If any traffic is seen entering the network with a source address of the defined internal network the packet should be dropped and logged.

```
access-list 100 deny ip host 72.16.1.54 host 72.16.1.54 log
```

This entry denies traffic sourced and destined to same IP address on a router. This reasoning is demonstrated by the popular Land Attack which may involve the chargen service on Cisco routers. This attack works by the attacker sending a spoofed packet to TCP/UDP port 19 of a router in which the chargen service is running. The malformed packet specifies the source and destination IP addresses being the same – the IP of the external router interface. Upon reception, the router's chargen service generates a new packet in response setting the destination IP address to that of the source IP address of the original packet. This results in the router generating traffic to its own external interface. The router uses all available resources to fulfill the chargen requests resulting in a Denial of Service (DoS) of legitimate traffic attempting to traverse the router.

```
access-list 100 deny ip 127.0.0.0 0.255.255.255 any log
access-list 100 deny ip 0.0.0.0 0.255.255.255 any log
```

According to RFC 1700, the entire 127-dot range is used for loopback addressing only. No traffic entering the network should have a source address of 127-dot. The same RFC specifies that the network 0.0.0.0/8 may be used to reference source hosts on “this” network only. No traffic with a zero-dot address is expected inbound and should be dropped.

```
access-list 100 deny ip 10.0.0.0 0.255.255.255 any log
access-list 100 deny ip 172.16.0.0 0.15.255.255 any log
access-list 100 deny ip 192.168.0.0 0.0.255.255 any log
```

There is no RFC that specifically forbids the practice of routing RFC 1918 and other private addresses on the Internet. RFC 3330 specifically points this out, “…the Internet does not inherently protect against abuse of these addresses; if you expect (for instance) that all packets from the 10.0.0.0/8 block originate within your subnet, all border routers should filter such packets that originate from elsewhere. Attacks have been mounted that depend on the unexpected use of some of these addresses.” Thus it is up to ISPs and customers to properly filter these addresses so that rogue packets don't propagate on the Internet. If packets with these source addresses are discovered coming from the Internet, then the packet is definitely suspicious and should be discarded.

```
access-list 100 deny ip 169.254.0.0 0.0.255.255 any log
access-list 100 deny ip 192.0.2.0 0.0.0.255 any log
```

Although RFC 3330 is defined as a memo to the Internet community, the NSA has included these ranges of addresses into those that should be restricted. RFC 3330 states that the 169.254.0.0 255.255.0.0 network is “the ‘link local’ block. It is allocated for communication between hosts on a single link. Hosts obtain these addresses by auto-configuration, such as when a DHCP server may not be found.” The same RFC comments that the network 192.0.2.0/24 is “assigned as ‘TEST-NET’ for use in documentation and example code. It is often used in conjunction with domain names example.com or example.net in vendor and protocol documentation. Addresses within this block should not appear on the public Internet.”

```
access-list 100 deny ip 224.0.0.0 15.255.255.255 any log
```

Depending if an organization is expecting multicast traffic from an untrusted source this ACL entry might not apply. Most organizations don't anticipate receiving multicast traffic from the Internet. Unless known otherwise, it is most likely safe to deny any multicast sourced traffic. This ACL also includes the experimental class “E” network as outlined in RFC 1700. Any traffic sourced with IPs from 240.0.0.0 to 255.255.255.254 should be denied.

```
access-list 100 deny ip any host 100.20.1.255 log
access-list 100 deny ip any host 100.20.1.0 log
access-list 100 deny ip host 255.255.255.255 any log
```

These ACL entries drop packets destined for the network or broadcast address of the network as well as packets with a source of a broadcast. Sending packets to a broadcast address is useful for attackers wishing to use a number of PCs for a Smurf-attack. This attack occurs when a spoofed packet is sent to the broadcast address of a subnet. All PCs on that subnet in turn respond to the spoofed IP address. The result is a potential DoS attack against the spoofed host. This practice is known as “amplifying,” thus the PCs on the subnet are known as Smurf Amplifiers. The network address is also specified due in part to some legacy systems that use the network address as a valid broadcast address.

```
access-list 100 permit tcp any 100.20.1.0 0.0.0.255 established
```

Because TCP is connection-oriented, all TCP sessions begin with a 3-way handshake to establish a known state between two communicating hosts. The initial packet sent to a destination host has the SYN bit set in the TCP header. The destination host will reply to the source host with the SYN and ACK bits set in that packet. To complete the handshake, the source host sends the final handshake packet with the ACK bit set. Once the session is established all subsequent packets for this session should have the ACK bit set acknowledging the delivery or setting a delivery checkpoint of TCP packets. The keyword, *established*, in this ACL checks to verify that the ACK and/or the RST bits are set. If the ACK bit is set, the assumption is that the TCP session has been already been initiated by an internal host. If the RST bit is set, the assumption is that an existing, valid TCP session is being torn down by the external host.

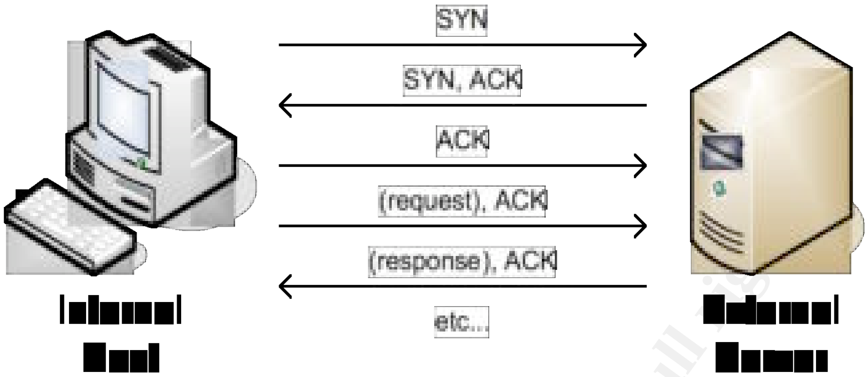


Figure 2

```
access-list 100 deny icmp any any echo log
access-list 100 deny icmp any any redirect log
access-list 100 deny icmp any any mask-request log
access-list 100 permit icmp any 100.20.1.0 0.0.0.255
```

This list of entries dealing with the ICMP protocol attempts to filter unwanted icmp traffic that is inbound to the network. ICMP echo requests are often used by scanners such as NMap to obtain information about the internal structure of a network. By using ICMP an attacker can map-out a network thereby giving him/her valuable information of where to plan their next move. ICMP-redirect requests can modify internal host route-tables which may result in route hijacking. Not all ICMP traffic is inherently bad. ICMP is intended to also provide information about the condition of the network such as when a packet is to large or when congestion occurs for a network segment along a packet's path to its destination. Allowing some inbound ICMP traffic gives the router and other devices important information about conditions of the external network. In addition it can provide useful to network administrators in troubleshooting networking issues involving external hosts.

```
access-list 100 deny tcp any any range 6000 6063 log
access-list 100 deny tcp any any eq 6667 log
access-list 100 deny tcp any any range 12345 12346 log
access-list 100 deny tcp any any eq 31337 log
access-list 100 deny udp any any eq 2049 log
access-list 100 deny udp any any eq 31337 log
```

These ACLs deal with various nefarious programs and their associated TCP port numbers. These programs, known as Zombies, can be associated with viruses, worms, Trojan horses, and other mal-ware. The NSA security guide provides a limited list of these port numbers, but an administrator should tailor this ACL to their own environment. A word of caution here is to limit large ranges of port numbers from being filtered when dealing with UDP based traffic. It is possible to inadvertently inflict a DoS on internal hosts due in part that internal host may be using an ephemeral port number in the range of the denied ports. If several ports are filtered, the host may mistakenly interpret a series of failed responses to be problems with the network. If it is possible to configure a firewall or the clients to exclude the same list of ports for communication, a difficult and confusing issue can be avoided.

```
access-list 100 permit tcp any gt 1023 100.20.1.0 0.0.0.255 eq ftp
access-list 100 permit tcp any gt 1023 100.20.1.0 0.0.0.255 range 1024 5000
```

According to RFC 959, there are two different methods of connectivity using FTP. The first of these is known as Active FTP. It uses two TCP ports for communication (20-data and 21-control). FTP communication is initiated on the Control Channel (typically to port 21). As part of the control information, the client specifies an ephemeral port number with which the server may establish data communications. The server then establishes a Data Channel (typically from port 20) to the client specified ephemeral port. If a firewall is installed anywhere between the client and server, the Data Channel communication will most likely fail because a stateful filter will typically deny traffic initiated externally unless specifically permitted. For this purpose there exists a second method of communication – Passive FTP.

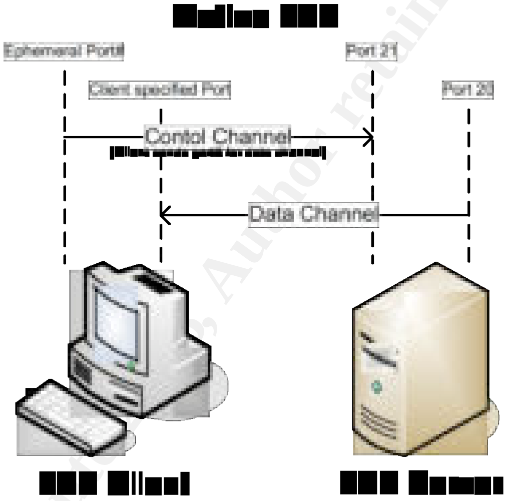


Figure 3

Passive FTP initiates FTP communication in the same manner as Active FTP. As part of the return communication, the server specifies an ephemeral port to be used for Data Channel communications. The FTP client connects to the server specified ephemeral port from a client ephemeral port to establish the Data Channel. If the client is initiating a Passive FTP session from inside a firewall, all returning packets from the FTP server will have the ACK bit set. The inbound packet will be permitted by the *tcp established* rule previously mentioned.

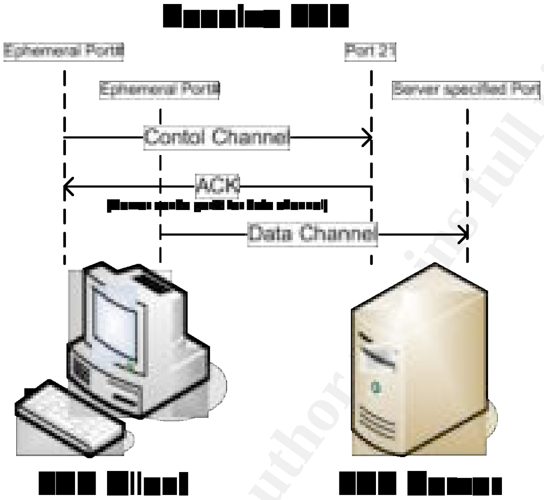


Figure 4

There is an issue faced when providing FTP services to an external client. Assuming the external client is also placed behind a firewall the access-list rules should be based on Passive FTP. The initial Control Channel connection will be to port 21 from an ephemeral port. The subsequent Data Channel communication will be from an ephemeral port on the external client to a server specified ephemeral port on the server.

According to Microsoft, the default allocated ports for a Microsoft IIS 6.0 FTP server running in passive mode is 1024 to 5000. The access-list needs to permit traffic from the FTP client ephemeral port to the FTP server specified ephemeral port in this range. Microsoft further instructs that this default port range can be changed by editing “the **PassivePortRange** property in the metabase.” Due to this range of port permitting traffic, it is advised that this rule be defined near the bottom of the access-list.

An alternate method of dealing with this issue is to utilize reflexive access-lists or implement the CBAC software on the router which can be obtained from Cisco Systems. These technologies implement stateful packet filtering. The discussions of these two technologies are beyond the scope of this paper.

```
access-list 100 permit tcp any gt 1023 100.20.1.0 0.0.0.255 eq www
access-list 100 permit tcp any gt 1023 100.20.1.0 0.0.0.255 eq smtp
access-list 100 permit tcp any gt 1023 100.20.1.0 0.0.0.255 eq 443
```

If a network is hosting its own web-site or mail server then these ACL entries are important. These entries allow the first packet of a TCP handshake to enter the network. All other packets will be processed by the *TCP established* rule specified earlier in the list.

```
access-list 100 deny udp any any range 33400 34400 log
```

This entry deals with various programs and the UDP port numbers they are typically associated with. Unix hosts typically use UDP ports 33400-34400 for the program traceroute. The associated rule denies inbound UDP requests associated with traceroute coming from external Unix hosts. The denial of these packets is akin to the denying of ICMP echo-request packets that are generated on Windows hosts.

```
access-list 100 permit udp any eq domain 100.20.1.0 0.0.0.255 gt 1023
access-list 100 permit udp any eq domain 100.20.1.0 0.0.0.255 eq domain
access-list 100 permit udp any gt 1023 100.20.1.0 0.0.0.255 gt domain
access-list 100 permit tcp any gt 1023 10.20.1.0.0.0.0.255 eq domain
access-list 100 permit tcp any eq domain 10.20.1.0 0.0.0.255 gt 1023
access-list 100 permit udp any eq ntp 100.20.1.0 0.0.0.255 eq ntp
access-list 100 permit udp any eq ntp 100.20.1.0 0.0.0.255 gt 1023
access-list 100 permit udp any gt 1023 100.20.1.0 0.0.0.255 eq ntp
```

UDP is a connectionless protocol. Each UDP packet is unique to itself so there exists no method of determining whether this packet is associated with any other UDP packets that are inbound. For each protocol allowed in there is a combination of source/destination ports that may be used. For an outbound packet the source port can be either an ephemeral port or the port of the service on which the service is communicating. The destination port for the outbound packet would be the same port in either case. The same is true on an inbound connection. Examining an inbound UDP packet for DNS, the source/destination port pairs can be one of three valid combinations (see the above ACL). For DNS based TCP traffic there also exists two additional entries. If a DNS packet exceeds 512-bytes of information, DNS indicates that more information will be forthcoming via a TCP connection. This one entry allows a DNS tcp connection to be initiated from an external DNS server inbound or from an internal DNS server outbound. This entry does not apply to every configuration.

```
access-list 100 deny tcp any range 0 65535 any range 0 65535 log
access-list 100 deny udp any range 0 65535 any range 0 65535 log
access-list 100 deny ip any any log
```

Finally, any traffic that has not met the explicit permit rules above is dropped and logged. As a policy, all denied traffic is logged for the purposes of forensic data and other uses, should it be needed. The router should be configured to report this data to at least one logging server.

Filtering ingress traffic is only half of the solution. A network administrator does not want information regarding his/her internal network to "leak" out to the Internet. In addition, a network administrator does not want the internal network to be the source for an attacker to launch their next series of attacks. Below is another access-list that filters outbound traffic.

```
access-list 102 remark This access-list filters egress traffic to an
untrusted destination
access-list 102 deny ip host 100.20.1.0 host 100.20.1.0 log
```

Mentioned before, remarks can enhance an ACL by providing information regarding its use.

```
access-list 102 permit icmp 100.20.1.0 0.0.0.255 any echo-reply
access-list 102 permit icmp 100.20.1.0 0.0.0.255 any parameter-problem
access-list 102 permit icmp 100.20.1.0 0.0.0.255 any packet-too-big
access-list 102 permit icmp 100.20.1.0 0.0.0.255 any source-quench
```

This is a set of rules that limit ICMP packets from escaping the network. ICMP is a useful protocol from attacker's standpoint. Assume that an attacker performs a TCP full-connection scan to map out the network. In the event that a system does not have a valid TCP port open, the system may respond with an ICMP packet indicating this fact. This response or lack thereof provides valuable information to the attacker. Not only does this information confirm to the attacker that the scanned TCP port is not available, but also confirms that a live host exists at the specified IP address which is responding to network traffic. The attacker could begin to perform additional scanning in attempts to gain access to the live system. The ICMP packets returning to the attacker may also provide enough information to determine what type of host operating system is generating the response (a.k.a Application Fingerprinting). Many security consortiums encourage limiting ICMP outbound packets. There are three types of ICMP that might be allowed: parameter-problem, packet-too-big and source-quench. According to the NSA security guide, "Echo-packet users will be able to ping external hosts. Parameter Problem and Source Quench packets improve connections by informing about problems with packet headers and by slowing down traffic when it is necessary. Packet Too Big is necessary for Path MTU discovery."

```
access-list 102 deny tcp any any range 1 19 log
access-list 102 deny tcp any any eq 43 log
access-list 102 deny tcp any any eq 93 log
access-list 102 deny tcp any any range 135 139 log
access-list 102 deny tcp any any eq 445 log
access-list 102 deny tcp any any range 512 518 log
access-list 102 deny tcp any any eq 540 log
```

This set of rules limit various TCP protocols. Among these are chargen, whois, NetBIOS, Microsoft-ds, uucpd and others. These services provide system information such as usernames, and services available on internal hosts. With a known username or service, an attacker can run a brute-force password attack against that host with the given username.

```
access-list 102 permit tcp 100.20.1.0 0.0.0.255 eq www any gt 1023
established
access-list 102 permit tcp 100.20.1.0 0.0.0.255 eq smtp any gt 1023
established
access-list 102 permit tcp 100.20.1.0 0.0.0.255 eq 443 any gt 1023
established
```

```
access-list 102 permit tcp 100.20.1.0 0.0.0.255 eq ftp any gt 1023
established
access-list 102 permit tcp 100.20.1.0 0.0.0.255 range 1024 5000 any gt 1023
```

established

This list belongs to the service hosting environment. An external host connecting to a hosted server will initiate a connection with a SYN packet. All subsequent TCP packets for this session will have the ACK bit set which will further protect the network by using the *established* keyword on each list entry.

```
access-list 102 permit tcp 100.20.1.0 0.0.0.255 gt 1023 any
```

This outbound rule permits any outbound tcp packet with a source ephemeral port to any destination.

```
access-list 102 permit udp 100.20.1.0 0.0.0.255 gt 1023 any eq domain
access-list 102 permit udp 100.20.1.0 0.0.0.255 eq domain any gt 1023
access-list 102 permit udp 100.20.1.0 0.0.0.255 eq domain any eq domain
access-list 102 permit udp 100.20.1.0 0.0.0.255 gt 1023 any eq ntp
access-list 102 permit udp 100.20.1.0 0.0.0.255 eq ntp any eq ntp
```

The list now contains rules to permit some UDP protocols. As said previously there are three combinations of source/destination port pairs that must be considered. NTP is not being hosted so one of those rules can be eliminated.

```
access-list 102 permit udp 100.20.1.0 0.0.0.255 any range 33400 34400 log
```

If there are Unix hosts on the internal network they might generate a different form of a “ping” packet. Unix hosts traditionally use UDP for the traceroute program instead of using an ICMP echo-request. If there aren’t any Unix hosts to worry about, omit this statement.

```
access-list 102 deny tcp any range 0 65535 any range 0 65535 log
access-list 102 deny udp any range 0 65535 any range 0 65535 log
access-list 102 deny ip any any log
```

To finish off the outbound rule-set, deny all remaining packets that didn’t match any of the rules above. Again, every deny rule specifies to log the match.

Implementation

With the filter designed, implement the filters on the external interface of the perimeter router using the following series of commands:

```
Router# config t
Router(config)# int s0/0
Router(config)# ip access-group 100 in
Router(config)# ip access-group 102 out
Router(config)# end
```

To make subsequent changes to an ACL, it is advised not to use the traditional method of removing the ACL with the "no" command followed by recreating the list with the same number or name. The issue raised in this situation is that for a time; however brief, some or all rules are removed thus lowering a layer of defense and exposing the internal network unnecessarily.

A superior method of making a change to an ACL is to create a completely new ACL, for example 101, followed by issuing the "*ip access-group 101 in*" command on the appropriate interface. In this instance, only the ACL associated with the interface is changed and no unnecessary exposure to the internal network occurs. Additionally, due to the implicit deny at the end of all Cisco based ACLs, creating a new list with the list already assigned to the interface may result with a DoS against the administrator themselves. This may occur if an administrator who manages the router being modified via the same port related to the list does not include an entry as the first entry in the list giving them continued access to the router. The router may deny further communication from the administrator's PC, after the first list entry is entered, thus inhibiting the administrator from adding the remainder of the list.

After the list associated with the interface is changed, the old access list should remain unmodified until the new list is working properly for a sustained time-frame. At some future point of time the old access-list can be deleted by using the "*no access-list 100*" command. Access-list 100 can again be used for the next change. Alternate between lists 100 and 101 any time a change is needed. This will provide a back-out plan that can be implemented quickly. Use the same methodology for the outbound access-list 102.

Verify the access-list is functioning properly by executing programs that result in permitted and denied traffic in/outbound of the network. By also using the following commands an administrator can determine if the ACLs are properly filtering the correct traffic.

```
Router# show access-list 100
Router# show access-list 102
```

These commands will show each line of the access-list and how many instances packets have matched each rule.

```
Router# show log
```

This command will display the log entry generated by any rule which includes the *log* keyword in the rule. Information generated by the router for a given rule specifies the date/time the rule was matched, access-list number, source/destination IP address, and source/destination port numbers. In conjunction with firewall logs, an administrator will be able to better identify patterns of suspicious traffic and the events associated with them.

Results

How well can static packet filters protect a network as the first line of defense? As an experiment, a Nessus vulnerability scan was performed against a Windows 2000 Server system (with no service packs installed) with externally available ftp, http, smtp and dns services.

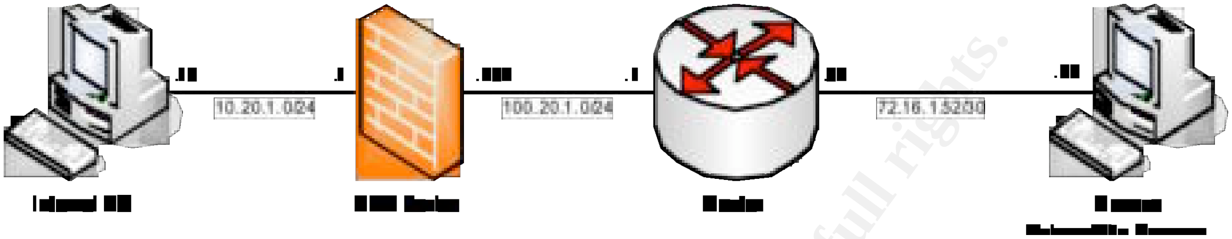


Figure 5

A Cisco 1760 router was placed between the Windows 2000 Server and the Nessus vulnerability scanner. The slowest link speed was 10Mbps in the entire topology. Initially, the Cisco router was not configured with any ACLs so all traffic was implicitly permitted.

After the initial scan concluded, the previously outlined NSA based access-list was associated to the external interface in- and out-bound. A second vulnerability scan was executed identical to the first. To get an accurate measurement of effectiveness, the access-list hits were recorded. To measure impact on the router, the CPU utilization was recorded as a function of the percent of the scan completed. These results combined with the results from the vulnerability scan are shown in the following figures.

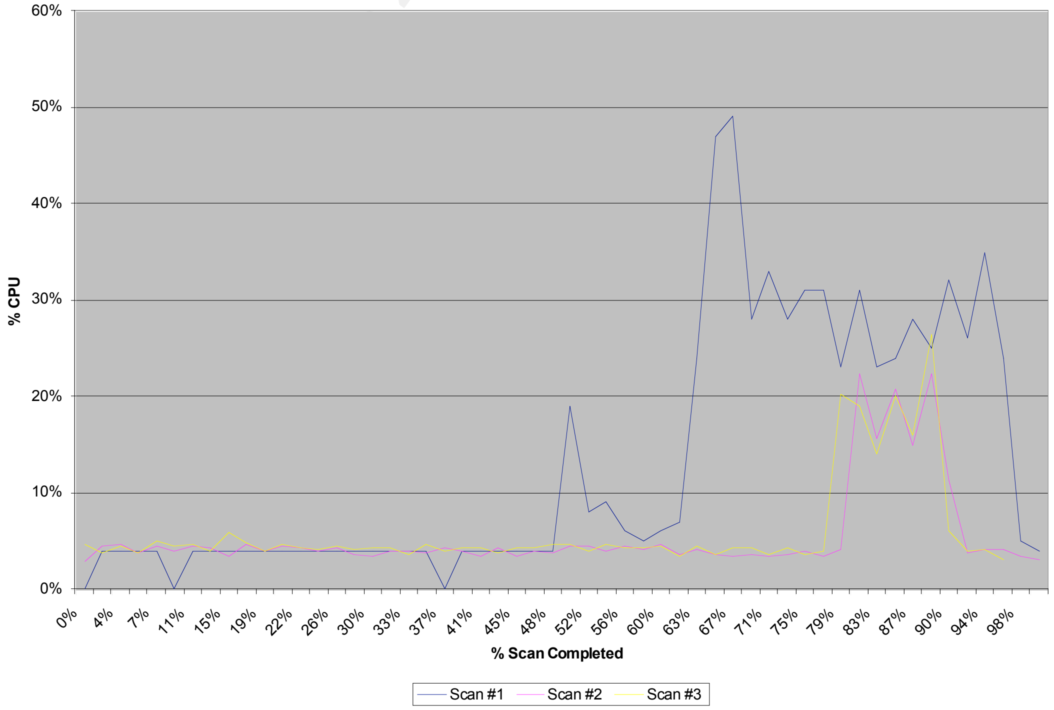
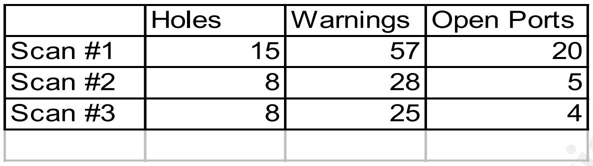


Figure 6

The first vulnerability scan completed in approximately 5 minutes time. There was a significant jump in CPU utilization at about 65% completion to the end of the scan as is shown in the above graph. Scan #2 completed in about 30 minutes time. A jump in CPU utilization didn't occur until about 80% completion of the scan and resulted in less overall CPU utilization than scan #1. A third scan was executed which results will be discussed later.

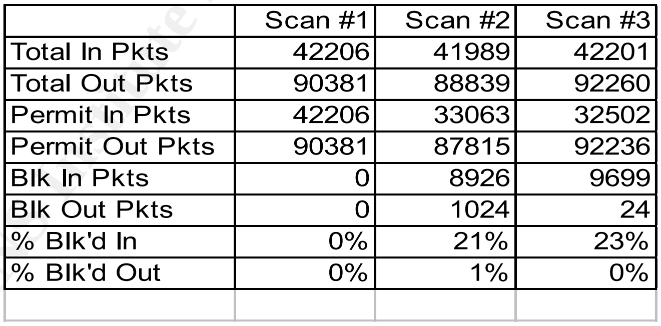


	Holes	Warnings	Open Ports
Scan #1	15	57	20
Scan #2	8	28	5
Scan #3	8	25	4

Figure 7

Examining the vulnerability results, scan #1 discovered 15 security vulnerabilities (Holes), 57 potential security threats (warnings) and 20 open ports on the system (available services). Scan #2 resulted in nearly a 50% reduction in the number of vulnerabilities and security threats and a 75% reduction in open ports available.

By examining the results of the *show access-list* command run after each scan a determination was made of how many packets were denied vs. permitted. Each scan resulted in approximately 42,000 packets inbound and 90,000 packets outbound. The filter-lists effectively blocked over 20% of the packets. This percentage is misleading as it only represents packets dropped as the result of a vulnerability scan. The packets associated with a vulnerability scan don't accurately represent the normal distribution of traffic inbound from the Internet. However, the percentages here do show the potential impact of a packet filter on inbound traffic.



	Scan #1	Scan #2	Scan #3
Total In Pkts	42206	41989	42201
Total Out Pkts	90381	88839	92260
Permit In Pkts	42206	33063	32502
Permit Out Pkts	90381	87815	92236
Blk In Pkts	0	8926	9699
Blk Out Pkts	0	1024	24
% Blk'd In	0%	21%	23%
% Blk'd Out	0%	1%	0%

Figure 8

In order to obtain more accurate data regarding the application of a filter, a similarly configured packet filter to the one used in this research was applied to a production router at a financial institution over an eight-hour period. The organization hosts a web-site that receives over 10 million hits per month and also services an organization of over 100 internal employees. Over the eight-hour period in which statistics were recorded the router blocked roughly 27% of the total traffic traversing the router. More importantly, the denied traffic was legitimately denied and not falsely denied. The log entries resulting from denied traffic were used to confirm this.

Mitigation

As previously stated, the recommended NSA access-list provides a balance between administrative overhead and principle of “least privilege.” The NSA ACL allows most outbound TCP traffic. It also allows only port level traffic inbound valid for the defined subnet. The list can be further refined to restrict all outbound traffic except for the traffic which is explicitly defined and inbound traffic defined to not only a port and subnet, but to a port and valid IP address as is shown below.

```
access-list 100 permit tcp any gt 1023 host 10.20.1.10 eq www
access-list 100 permit tcp any gt 1023 host 10.20.1.10 eq ftp
access-list 100 permit tcp any gt 1023 host 10.20.1.10 eq smtp
access-list 100 permit tcp any gt 1023 host 10.20.1.10 eq 443
access-list 100 permit udp any gt 1023 host 10.20.1.10 eq domain
access-list 100 permit tcp any eq ftp-data host 10.20.1.10 gt 1023
access-list 100 permit udp any eq domain host 10.20.1.10 eq domain
access-list 100 permit udp any eq domain host 10.20.1.10 gt 1023
access-list 100 permit udp any eq ntp host 10.20.1.10 eq ntp
access-list 100 permit udp any eq ntp host 10.20.1.10 gt 1023

access-list 102 permit tcp host 10.20.1.10 eq ftp any gt 1023
access-list 102 permit tcp host 10.20.1.10 eq www any gt 1023
access-list 102 permit tcp host 10.20.1.10 eq smtp any gt 1023
access-list 102 permit tcp host 10.20.1.10 eq 443 any gt 1023
access-list 102 permit tcp host 10.20.1.10 gt 1023 any eq www
access-list 102 permit tcp host 10.20.1.10 gt 1023 any eq 443
access-list 102 permit tcp host 10.20.1.10 gt 1023 any eq ftp
access-list 102 permit udp host 10.20.1.10 gt 1023 any eq domain
access-list 102 permit udp host 10.20.1.10 eq domain any gt 1023
access-list 102 permit udp host 10.20.1.10 eq domain any eq domain
access-list 102 permit udp host 10.20.1.10 gt 1023 any eq ntp
access-list 102 permit udp host 10.20.1.10 eq ntp any eq ntp
```

In addition, any inbound TCP traffic with the ACK bit set will be allowed into the network. According to Cisco, “A stateless IP packet filter…must make all of its forwarding decisions for any specific packet based only on information in that packet. If the filtering is based on criteria such as TCP or UDP port numbers, the necessary information is typically present only in the initial fragment of a fragmented datagram. It is therefore impossible to tell if a non-initial fragment is part of a forbidden datagram or of a permitted one. Therefore, stateless packet filters that use such criteria must pass all, or substantially all, non-initial fragments. Such filters rely on blocking of initial fragments to prevent completed delivery of any forbidden packets. This makes them vulnerable to the fragmentation denial of service attacks...” Many attackers will utilize packet crafting tools in which the ACK bit can be artificially set to bypass the ACK/RST filter.

The changes described below will not prevent an attacker from attempting to circumvent this rule; however, the potential for a breach is will be reduced as the inbound filtering rules will be more strict.

```
access-list 100 permit tcp any eq www host 10.20.1.10 gt 1023 established
access-list 100 permit tcp any eq 443 host 10.20.1.10 gt 1023 established
access-list 100 permit tcp any eq ftp host 10.20.1.10 gt 1023 established
access-list 100 permit tcp any eq telnet host 10.20.1.10 gt 1023 established
```

The third scan mentioned previously was completed after implementing the additional security suggestions above. Scan #3 didn't show any significant deviation from Scan #2; however, as previously mentioned the vulnerability scan does not accurately depict normal network traffic on the Internet. This would coincide with an observation made by Ray Kaplan in Information Security magazine. Quoting from his article "Safe at Any Speed," Ray states, "The primary performance hit comes when activating ACLs in the first place; once activated, there's little overhead associated with adding ACLs, but only up to a point..."

The additional configuration items did require further testing. Any change to external network access required modification and retesting of the ACL, proving the increased administrative burden described previously.

As RFC 3330 indicates, perimeter routers should filter IP addresses based on RFC 1918 and other private ranges. To prevent any traffic from an organization that might be sourced from a private range, spoofed or not, a router administrator might consider adding the same list of filters that initially deny inbound traffic from the given list of private IP addresses described in the Analysis section of this paper.

The NSA and other security resources recommend using ACLs to further enhance security with regards to routing updates, direct connections to the router via telnet or SSH, NTP updates, SNMP traps and management, remote access connectivity, and others.

Changing the FTP Passive port range to limit the number of potential inbound ports is encouraged by Microsoft. A restriction of ports associated with Passive FTP in the ACL will limit the number of potential weaknesses at the network perimeter.

Additional security features can be included by way of packet filtering ACLs on Cisco routers. Some of those items that are available out-of-the-box with any IOS-based Cisco routers are:

- Reflexive – Introduce state to communication sessions.
- Dynamic – Change filters based on dependencies of other rules.
- Time based – Communication that can take place at specific times or for durations of time.
- User based – Sessions that are based upon a particular user or group authentication.

Software that can be purchased from Cisco and added to a router includes:

- CBAC (Cisco Based Access Control) – Cisco's IOS based firewall feature set which implements a fully featured stateful filter on a router and can be used to prevent additional attacks.
- IDS – Intrusion Detection System which can detect anomalous network behavior and change ACLs to block that behavior.

The trade-offs for providing this level of security at the perimeter are additional CPU and memory resources needed by the router, additional network latency, a more expensive router as well as an additional administrative burden placed upon the router administrator. Purchasing these additional features for a branch or small office might prove beneficial due to a lower overall cost, however proceed with caution; the potential security ramifications could be high. Utilizing a single platform and technology for network security is not consistent with defense-in-depth strategies. Should the system become compromised, there won't be any systems to protect the internal network.

Conclusion

In summary this paper described that some reasons for implementing a static filter list at a network perimeter is for filtering Internet noise, providing an additional layer of security for an organization, and for providing a minimal level of security at locations that prove unfeasible to place a stateful firewall. A filter list can improve or augment the overall security of an organization by providing yet another layer of security between an organization's corporate network and the Internet. This additional layer could be implemented with little cost and a minimal impact to network resources.

A recommended approach to this implementation is described in the paper by the NSA, "The NSA Router Configuration Security Guide." The paper discusses each rule presented in the ACL and explains its purpose and how it can enhance security. Using the NSA guide as a template, an ACL was conceived and implemented in a lab network. Commands to properly implement, change, and verify the function of the ACL was given.

The ACL was tested with the use of a vulnerability scanner and then in a production environment connected to the Internet. The results from these tests demonstrated the usefulness of a packet filter in regards to network security. Additional suggestions to further enhance security of the NSA filter-lists were discussed. These modifications were also tested and demonstrated little improved security in the test environment, but it was concluded that in a hostile networking environment like the Internet, the modifications would indeed provide additional security benefits. Other enhancements to Cisco ACLs were presented that could provide stronger security measures if warranted.

References

Cisco Systems. “Cisco Security Advisory: Cisco PIX and CBAC Fragmentation Attack.” Document ID: 23885. Revision 1.2. 11 September 1998. 10 August 2004 http://www.cisco.com/warp/public/770/nifrag.shtml

Cisco Systems. “Evolution of the Firewall Industry.” 28 September 2002. 10 August 2004 http://www.cisco.com/univercd/cc/td/doc/product/iaabu/centri4/user/scf4ch3.htm.

Coreth Consulting, Inc. "Firewalls vs. Packet Filters." 1999. 28 July 2004 http://www.coreth.com/techdocs/fw_vs_pf.html.

IANA. “Special-Use IPv4 Addresses.” RFC 3330. The Internet Society. September 2002. 5 August 2004 http://www.ietf.org/rfc/rfc3330.txt.

Kaplan, Ray. “Safe at Any Speed?” Information Security July 2002.

Kreiling, Janet. “Security for the Branch or Small Office.” Packet: Cisco Systems Users Magazine 4th Quarter 2003: 40 – 43.

Microsoft. “Configuring FTP Site Properties.” 27 February 2004. 10 August 2004 http://www.microsoft.com/resources/documentation/IIS/6/all/techref/en-us/iisRG_CFG_18.mspx?pf=true

Postel and Reynolds. “Assigned Numbers.” RFC 1700. October 1994. 6 August 2004 http://www.ietf.org/rfc/rfc1700.txt.

Postel and Reynolds. “File Transfer Protocol.” RFC 959. October 1985. 9 August 2004 http://www.ietf.org/rfc/rfc959.txt.

Rekhter, Moskowitz, Karrenberg, Groot, and Lear. “Address Allocation for Private Internets.” RFC 1918. February 1996. 5 August 2004 http://www.ietf.org/rfc/rfc1918.txt.

SCO Group. “Configuring Packet Filters.” 22 April 2004. 1 August 2004 http://ou800doc.caldera.com/en/NET_tcpip/filterN.intro.html.

Sedayao, Jeff. Cisco IOS Access Lists. O’Reilley, June 2001.

United States. National Security Agency. Router Security Configuration Guide. By Vanessa Antoine, Raymond Bongiorni, Anthony Borza, Patricia Bosmajian, Daniel Duesterhaus, Michael Dransfield, Brian Eppinger, Kevin Gallicchio, Stephen Hamilton, James Houser, Andrew Kim, Phyllis Lee, Tom Miller, David Opitz, Florence Richburg, Michael Wiacek, Mark Wilson, and Neal Ziring. 5 December 2003. 28 July 2004 http://www.nsa.gov/snac/routers/cisco_scg-1.1b.pdf.