Global Information Assurance Certification Paper

Firewalls and Perimeter Protection Practical Assignment

The headlined filtering items below will be examined on our Perimeter Defense System, which consists of a Cisco 2600 series router and a Mandrake – Linux 7.0 system. The Linux kernel version is 2.2.15 - packet filtering and IP forwarding options are built into the kernel. The Service Provider maintains the Cisco router; any request for setup is routed through the SP – therefore screen dumps of the router session are not available.

It cannot be stressed enough, knowing how the common protocols, services, and programs work will go a long way to enabling a secure network to be setup. Simply reading a book or going to a seminar cannot provide enough knowledge to implement any successful perimeter defense solution. Keeping up to date with the latest exploits and hacks is the best way to make sure that you are up to date on potential problems.

Inventorying services and then minimizing service availibilty to the outside world is one of the first steps to protecting the internal network from outside attacks. The more openings, the more likely an attacker is to find a flaw or bug in a service being supported through the web. Services not available to connect to, offer little to attackers even if the firewall does let them through. The perimeter solution should drop any information not specifically authorized to exit the facility as defined in the local or corporate security policy.

The actual IP addresses, services, and domain names have been modified to protect the somewhat innocent.

Block "spoofed" addresses-- packets coming from outside your company sourced from internal addresses or private (RFC1918 and network 127) addresses. Also block source routed packets.

Spoofed addresses are those that are made to look like packets from the internal network. They are created to sidestep security systems. In most networks, there should never be traffic coming in from the Internet with addressing that looks like it came from internal hosts. The exception to this could be a VPN depending on how it is configured. Most importantly traffic from private network addresses should never enter the network, this traffic should be dropped before reaching the firewall or internal network. It is also possible to receive packets that have been directed through a network based on routing information given at the source. This source based routing is almost certainly packets that are intended to avoid

security which do not belong in the network and should be dropped before entering.

Spoofing affects any service that relies on IP address authentication including RPC, the X Window services, and the R services. Spoofing can be used to fool a host into responding to another host creating a loop effect where bandwidth is consumed resulting in a denial of service.

From the privileged mode – enter **config t** then enter **no ip source-routing**, this will eliminate source routing through the router and all interfaces.

The internal network uses the 172.22.0.0/24 address range; the following item will block attempts to spoof internal addresses. Using access-lists at the router and internal private addresses, protection can be achieved without causing increased load on the firewall. Cisco routers have the ability to intercept packets and perform pattern matches on them. Using the following commands a standard access-list can be added to protect against this threat. A standard access list is only able to check the source addressing information, this enables faster pattern matching than other forms of ACLs.

This will create the anti-spoofing access control list on the router. The deny statements can be in any order, but the permit any should be at the end. Cisco Access Lists by default deny anything that is not explicitly stated in the ACL. [From privileged mode – enter terminal configuration (use **config t**)]

```
access-list 1 deny 192.168.0.0 0.0.255.255
access-list 1 deny 172.16.0.0 0.15.255.255
access-list 1 deny 10.0.0.0 0.255.255.255
access-list 1 permit any
```

The syntax for these access-lists is:

Access-list {number} [action] IP Address netmask in inverted form

Where {number} is the access-list number, this also correlates to the access-group number when specifying which interface to apply the access-list to. [action] is either deny or permit the data. IP Address. Inverted netmask refers to the range that should be matched up for a given address. A class C subnet looks like 0.255.255.255 in inverted form.

(For the interface that inbound traffic enters from the Internet – here Serial 0/0) This will apply the ACL to the Serial 0/0 interface to traffic entering the interface.

```
interface serial 0/0
ip access-group 1 in
```

To verify that you have setup the access-list on the desired interface and direction, use the show interface command. It should list the serial 0/0 interface as having the access-group number and direction to apply the access-group. (Here: ip access-group 1 in).

The same type of ACL should be setup for traffic exiting the router. This is called an Egress filter; it makes sure that only legitimate traffic is bound for the Internet – in effect preventing the network from littering on the Internet.

Login services-- telnet (23/tcp), SSH (22/tcp), FTP (21/tcp), NetBIOS (139/tcp), rlogin et al (512/tcp through 514/tcp)

Services such as telnet, ftp, rlogin, and NetBIOS are helpful for remote administration. This convenience can come at a cost. Telnet services typically operate on TCP port 23 and not encrypted sessions making user ids and passwords easy to capture since they are often in clear text. In comparison, rlogin services allow a user to conduct a remote session on a host – but rlogin can be configured to trust a host for password-less access! Not a good idea in this day and age.

A Secure Shell (ssh) is an alternative to telnet and rlogin that provides much stronger user authentication and several levels of encryption to protect data being transmitted. Several encryptions are available and also support for secure copying of files (http://www.ssh.fi for more information). The key to using ssh is choosing a strong pass phrase and using the latest patches. You'll need to have any client software capable of supporting the encryption you are using. 'ssh' can be used to forward ports to alternate addresses and ports and therefore provide access to a variety of other services not just terminal sessions. From a perimeter defense perspective it is always good to drop any packets destined for ports that are not known when using remote control type access ports. For this reason, specific drop filters are setup for many of the services not related to http traffic.

NetBIOS is the file sharing portion of Microsoft Networking and not surprisingly several exploits exist. The recommended course of action is not to allow NetBIOS traffic into or out of the firewall. You should not be doing file sharing across the Internet without the use of a VPN, if even then. To protect against NetBIOS scans and connection attempts, dropping TCP and UDP ports 135 – 139 will provide the necessary protection (See NetBIOS portion below for more information).

FTP access, client and server, can be exploited very easily and used to bypass a perfectly good firewall. In the case of bypassing a firewall, a session can be opened where a file can be downloaded and executed on a system without the user or administrator being aware. This service is disabled in our security policy except at the administrator's system.

Our firewall drops any attempt to connect to ports below 1024, unless explicitly stated in the rulesets. 'ssh' access is available only from the administrator's system, all others are dropped. The same is true for ftp access; no one is allowed ftp access except the administrator.

Near the end, the firewall rules will be listed and explained – noting the order in which they should be applied along with any pertinent information.

RPC and NFS—Portmap/rpcbind (111/tcp and 111/udp), NFS (2049/tcp and 2049/udp), lockd (4045/tcp and 4045/udp)

Remote Procedure Calls (RPC) runs a service called portmap or rpcbind, depending on the OS in question. This service acts a registry of services to ports for a given machine and can be used as an aid for a would-be attacker. This can reveal details of other services running on a system including nfs and lockd ports. Several RPC flaws have been exploited which cause systems to hang or service to become locked up. Routinely access to TCP/UDP port 111 is blocked in order to protect systems running rpcbind or portmap. Be advised that some Operating Systems may be accepting RPC inquiries on ports above 111 (eg. Solaris 2.x listens above port 32770)

Network File System (NFS) exports local files systems in a manner that enables access to remote hosts. Various exploits have been found for this service and while use of this service internally may be beneficial, external use should be

minimized. Care must be used when exporting home directories as security tokens and special encryption settings may be exported unintentionally.

For this network no NFS is being used and therefore the corresponding ports are dropped before passing through the firewall.

These services are blocked by the firewall, RPC information is not routed from the outside to the inside or vice-versa. Please see the filter description near the end of this paper to view the rules used to block access to portmap,nfs, and lockd.

More information available at:
http://www.sans.org/newlook/resources/IDFAQ/blocking.htm
http://www.cert.org/advisories/CA-94.15.NFS.Vulnerabilities.html
http://archives.neohapsis.com/archives/bugtraq/2000-06/0073.html

NetBIOS in Windows NT -- 135 (tcp and udp), 137 (udp), 138 (udp), 139 (tcp). Windows 2000 – earlier ports plus 445(tcp and udp)

The Windows family of operating systems has become very popular; this coupled with the less than vigilant security interest from Microsoft makes for a highly targeted platform. Briefly NetBIOS is used by Windows to provide connectivity to file systems and printers across various OS versions. Peer-to-Peer networking is the predominant model in use on most Microsoft networks; with more Windows NT and Windows 2000 being setup to address security concerns not addressed in previous Windows' releases. Information ranging from the system name, user id logged in, and the shares available are visible just by requesting the information. Exploits have been developed to hack passwords and freeze Windows systems using the NetBIOS interface, these ports are described further here.

Port 135 is Microsoft's version of portmap or RPC service locator; several services make use of this RPC locator including WINS, DHCP, and DNS. Several exploits exist making use of port 135. Microsoft's RPC service locator or endpoint mapper indexes pipes as well as services. Pipe information can be used to create multiple idle sessions and send CPU usage to near 100%. A default Windows installation will result in several open ports above 1024 which are dynamically assigned, endpoint mapper provides the information relating to which service has this port open. Traffic to this port should not be originating from outside the network and if so that traffic should be dropped.

The exception to this would be for services that require connecting multiple sites running MS products like Exchange server or SQL Server and need access to port 135. This service can be provided through alternate methods.

Special consideration should be given to port 593, which provides endpoint mapper services for RPC over HTTP for use with IIS or Exchange server. This should be viewed with the same level of concern as port 135. DCOM applications make use of this port as well for endpoint mapping. This is in effect an encapsulated RPC service being tunneled over HTTP.

Port 137 is referred to as the NetBIOS name service. At the Firewall frequently there are large numbers of incoming packets to port 137; due to the behavior of Windows servers that use NetBIOS (as well as DNS) to resolve IP addresses to names using the "gethostbyaddr()" function. As users surf Windows-based web sites, those servers will frequently respond with NetBIOS lookups; creating several port 137 connections.

Windows makes use of port 138 as a NetBIOS datagram service used for broadcasting information and maintaining the browse list a.k.a. Network Neighborhood. The main danger from traffic external to the network is that by crafting special messages sent to this port, a hacker can convince Windows that their machine is "local", and can therefore bypass some of Microsoft's security settings that differentiate between "local" and "internet" zones. Thereby defeating security settings in Internet Explorer and other MS Applications. SAMBA also makes use of this port for this service.

NetBIOS functions on port 139 for all file and print sharing activities. This is arguably the most dangerous port on the Internet today. This port facilitates the actual file sharing for Windows based systems and typically is the first port that is scanned for. Tight security policies and a filter at the firewall will go a long way to protecting against this port being taken advantage of.

Older Windows systems prior to Windows 2000 used ports 137,138, and 139 for Server Message Block (SMB). SMB is the protocol behind the file sharing on Windows platforms. Windows 2000 moves SMB communications to port 445. As before stated traffic outside the network should be dropped except where explicitly desired, the case of Active Directory spanning multiple sites or other MS products, which rely on replicated information and must use NetBIOS.

Our security policy forbids local users from having files or printers shared locally. Further we block ports 135 – 139 and 445 tcp and udp just to ensure that no

information is entering or leaving to or from those ports. Check the firewall ruleset near the end for the filter rules.

X Windows -- 6000/tcp through 6255/tcp

X Windows is at its lowest level a communication protocol. This protocol is multi-platform and available for most operating systems. X Windows utilizes a Client-Server method of network communication. This method allows a user to run a program in one location, but control it from a different location. The client is the system running the application remotely and the server is the system where the user sits and controls the program. Most X clients allow connections based on host validation – or IP Addresses. Token keys can also be used and a slightly more secure. Once a session is in progress, a spoofing attack could result in connection hijacking.

This network uses internal X access but does not allow traffic to exit the firewall and only specific internal addresses can use the X terminals. This is configured in the xhost program by typing: **xhost +mypc.domain.com** or simply the ip address. A (-) will remove the host if necessary.

Going back to the anti-spoofing ACL at the router, no spoofed traffic is allowed in and at the firewall we block ports 6000 through 6255. Thereby protecting our X Window services from the outside world and restricting access internally by using the xhost program to control connections.

Naming services-- DNS (53/udp) to all machines which are not DNS servers, DNS zone transfers (53/tcp) except from external secondaries, LDAP (389/tcp and 389/udp)

Domain Name Service, DNS, is the key building block for the Internet resolving human friendly address names to machine necessary number addressing. DNS is a virtual card catalog of Internet sites. Early DNS servers were found to be vulnerable to an exploit that allowed hackers to poison this lookup service by submitting a query for an address and providing answers as well that were not

correct. This information was cached at the server and used when a client would query for an address. Infamous cache poisoning of Hillary Clinton's website - http://www.thestandard.com/article/display/0,1151,5578,00.html for more information.

DNS used to provide lookup information for sites might also contain detailed information on internal sites when using DNS for internal name resolution as well as external name resolution. This information would detail the names and IP addresses of internal hosts, and depending on the naming scheme reveal the services provided by these systems. This detailed map in DNS is known as a zone map, often times a backup DNS server is setup for fault tolerance. This secondary DNS may be configured to receive updates from the primary DNS server, these updates are known as zone transfers. This ensures that the secondary has a fresh copy of the zone map. This zone map is very helpful to would-be hackers trying to perform reconnaissance on a targeted site.

DNS uses port 53 as the standard port to perform communication between clients and servers and servers to servers. Typically the UDP port 53 is used to resolve requests that are less than 512 bytes. DNS information larger than this is switched to TCP port 53. Zone transfers occur on TCP port 53.

Clearly poisoning of the DNS cache and being able to perform zone transfers can cause major problems for a network connected to the Internet. Several techniques are available that offer protection. The most important is the zone transfers. Our DNS setup provides for external DNS lookups through a third party. We use our own internal DNS servers that resolve IP addresses for the internal hosts. Our external DNS server provides resolution for email and our web site. That ensures that whatever security measures our provider uses, that zone maps from the external DNS server will prove no use to attackers. Our internal DNS server is not allowed to zone transfer with any other DNS server except it's backup. Specifically telling the primary DNS server to only zone transfer with the secondary DNS ensures that no unauthorized transfers will occur. The secondary has zone transfers disabled. Remember that the anti-spoofing ACL prevents spoofing of the secondary or primary DNS.

TCP port 53 is disabled through the firewall. UDP port 53 is only allowed to the firewall box, which is running masquerading. The firewall only allows connections to the DNS outbound, attempts to connect through port 53 inbound are dropped.

Mail—SMTP (25/tcp) to all machines, which are not external mail relays, POP (109/tcp and 110/tcp), IMAP (143/tcp)

Simple Mail Transfer Protocol, SMTP, is the protocol that transfers mail on the Internet. SMTP servers deliver mail to their destination. Often times an SMTP server will receive mail for another domain. How this mail is handled determines whether the server is configured for relaying. If the server passes the email on to the domain referenced, then the SMTP server is relaying the email. If the server returns or does not accept the email then it is not configured as a relay. Spamming occurs when emails are addressed to SMTP relay servers and the SMTP servers relay the information to the intended domain. This can result in increased processing by the mail server and also hides the tracks of those sending the SPAM. Generally port 25 should be connected to by the domain in which it is serving mail for, i.e. bob@ micro.com connects to micro.com SMTP mail server when sending his mail.

Spammers are constantly scanning for relay servers. They are looking for responses to port 25, being sure to disable relaying will prevent your site from being used as a relay post for SPAM. Further, dropping requests to connect to hosts that do not offer SMTP services is always a good idea. This network has an external SMTP mail server, in this case port 25 is open for outgoing SMTP but blocked for incoming SMTP.

Post Office Protocol (POP) was designed for offline mail viewing where a clients connects to a server and downloads mail for local processing. Port 109 has predominately been replaced with port 110 (POP2 to POP3) for mail. POP3 is the latest protocol that handles incoming mail from the server. Most often attackers are looking for POP3 or even POP2 service that can be broken into to gather emails. If a username is known then brute force password cracking is often attempted. Several flaws have enabled POP3 to become vulnerable to serious attacks. Some simply render the POP3 unusable, while others attempt to gain root access by circumventing the mail system. Inward POP2/3 scans should be dropped. External POP connections should be controlled. This network allows access to the external POP server but only on port 110 and only the designated IP Address. No internal connects are allowed to port 110 or 109.

Internet Message Access Protocol (IMAP) was designed for online and offline processing of mail and folders an improvement to POP. The issue here is obtaining root level access by compromising the security or exploiting a flaw in

the IMAP service. IMAP is not used at this network; therefore the port is blocked at the firewall.

Web—HTTP (80/tcp) and SSL (443/tcp) except to external Web servers, may also want to block common high-order HTTP port choices (8000/tcp, 8080/tcp, 8888/tcp, etc.)

Access to the Internet lately has translated into accessing websites via port 80 and port 443 for secure transactions. Typically these services are provided offsite, except in the case of new web based management of equipment – namely network equipment and servers. Web traffic should be outbound to the Internet on port 80 and should never be inbound on port 80. The same is true for port 443 and any of the higher-order HTTP ports. The firewall blocks incoming connection attempts to connect to ports 80, 443, 8000, 8080, 8888, and 8989. As new equipment that supports remote management is installed, the management ports are configured to be included in the range that is blocked.

Devices that are running remote administration can be exploited or information gathered that might aid in the hacking of a network. HTTP traffic can be used to attack hosts in the same manner as viruses in email.

"Small Services"-- ports below 20/tcp and 20/udp, time (37/tcp and 37/udp)

“Small Services” are actually daemons running on various hosts by default; Unix boxes and Cisco routers are examples of this. Mostly these are tools for diagnostic tests, but with any tool they can be used for malicious activities. Attacking these services can cause a system to crash or become unstable, 100% CPU utilization is also possible. For the Cisco router, the commands below will disable these services on the router:

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

Finally we want to block these ports from being routed to internal or external hosts. This is accomplished at the firewall.

Miscellaneous-- TFTP (69/udp), finger (79/tcp), NNTP (119/tcp), NTP (123/tcp), LPD (515/tcp), syslog (514/udp), SNMP (161/tcp and 161/udp, 162/tcp and 162/udp), BGP (179/tcp), SOCKS (1080/tcp)

Trivial File Transfer Protocol (TFTP) provides basic file transmission capabilities with no user authentication. If it is being used on a network it should be configured with the most restrictive setup possible since it offers read and write access to a system running TFTP.

Finger service provides extensive information about a given user. It can easily be used in malicious attempts. Internal usage can be beneficial; however, external access to finger service should be limited or eliminated where possible.

Network News Transport Protocol (NNTP) provides news feed services such as USENET. Access to this service for hackers can be to anonymously read and create messages and therefore should be moderated.

Network Time Protocol (NTP) provides a means to synchronize systems with a known source. This service can be exploited to offer incorrect times, hang systems, and on some versions gain super user access.

Line Printer Daemon (LPD) provides distributed printing across a network. External access to this service can result in printers being hung or systems locking up.

Syslogd service is used for logging of events either locally on from network devices configured to do so. Syslogd connection attempts from outside may be trying to crash syslogd. On most Unix based systems crashing syslogd will hang or freeze the system.

Simple Network Management Protocol (SNMP) is the protocol of choice for managing devices that make up the Internet (routers, switches, etc). Access is controlled through the use of noun names that act as community names. These community names either allow read-only or read-write access and do not provide user authentication. For this reason it is recommended to create unique community names. Access to SNMP can be achieved through brute force password guessing; therefore access to SNMP services should be tightly

controlled. Some devices allow use of access controls to restrict access to SNMP services.

Border Gateway Protocol (BGP) is used to route packets between autonomous systems (An autonomous system is a set of routers that operate under the same administration.) Often updates are received that are misconfigured and or misbehaving, if BGP isn't being used for routing tables it should be dropped.

SOCKS is used for tunneling through a firewall from internal addresses to service on the Internet. Port scanning is often attempted to determine if SOCKS is being used, and if found attempts are made to gain access to the SOCKS tunnel. If successful this would open a door into the heart of a network, straight through the firewall.

Using the IP Chains packet filtering, as described in detail later, the approach to security at this network has been to deny all and permit the exceptions. Keeping this in mind, all of these services are blocked from entering or leaving the firewall. Secondly only syslogd is allowed to receive connections from specified devices for log updates. SNMP is guarded with mixed character community names that require brute force processing to crack. TFTP is used for a specified location, and is manually enabled, for updating CISCO IOS.

ICMP-- block incoming echo request (ping and Windows traceroute), block outgoing echo replies, time exceeded, and destination unreachable messages except "packet too big" messages (type 3, code 4). (This item assumes that you are willing to forego the legitimate uses of ICMP echo request in order to block some known malicious uses.)

ICMP requests of any nature can be used to gain information about a host or group of hosts. It can be used to cause a router to divulge information that would ordinarily not be public and also can be used to poison a router's cache of routes. ICMP is used to aid in control of information flow and provide tools to test for connectivity. The issue at heart is the use of these tools by outsiders to conduct reconnaissance of networks.

The firewall blocks all ICMP activity before it enters the firewall or even the LAN. This may not be possible on all networks, but this network does not permit ICMP traffic to enter or leave.

Security measures:

As mentioned before the Cisco router has access-lists configured to block private addresses from entering the network. Further source routing is blocked at the router. These options are available at IOS level 10.3.9 and above.

IP Chains:
The IP Chains script looks like this –

```
#Variable to allow for upgrade to ip_tables
IPC="/sbin/ipchains"

#Flush old rules, delete the firewall chain if it exists
$IPC –F input
$IPC –F output
$IPC –F forward
$IPC -F firewall
$IPC -X firewall

#Set up the firewall chain and all chains to deny by default except for forward chain
$IPC -N firewall
$IPC -A firewall –j DENY -l
$IPC –P input DENY
$IPC –P forward ACCEPT

#Accept ourselves on the loopback device
$IPC -A input –i lo-s 127.0.0.1/32 -d 127.0.0.1/32 -j ACCEPT

#Setup Masquerading for the range spcified
#
$IPC -A forward -s 172.22.0.0/24 -j MASQ
echo 1 > /proc/sys/net/ipv4/ip_forward

#Accept DNS
$IPC -A input -p udp –i eth0 –d 200.200.200.5 --source-port 53 -j ACCEPT

#Accept SSH. This line is modified if remote access is desired
# Only Admin remote system is permitted access
#$IPC -A input -p tcp –s 207.207.208.1/32 --destination-port 22 -j ACCEPT

#Accept outbound mail connections to known mail server
$IPC -A input -p tcp –d smtp.domain.com --destination-port 25 -j ACCEPT

#Accept outbound HTTPd Requests.
$IPC -A input -p tcp  --destination-port 80 -j ACCEPT –d ! 11.22.33.44/32

#Send everything else to the firewall chain.
$IPC -A input -p icmp -j firewall
$IPC -A input -p tcp --syn -j firewall
```

```
$IPC -A input -p udp -j firewall
```

Initially, all the chain rules are flushed and set to deny.
This enables the policy of denying everything that is not specified.
We are using the firewall chain to deny and log all the other packets.
We enable the localhost to talk to itself, with the next section.
The MSAQ command enables Linux to hide a subnet behind a given IP for accessing external sites. Enabling forwarding ensures that masquerading is able to route the packets in and out. The external sites see only the firewall's ip address. DNS requests are only viable when directed at the outsourced DNS server on the UDP port 53. No zone transfers or TCP based DNS connections are allowed. Provisions are made for accessing the firewall remotely with the commented line regarding SSH. This line is uncommented and the script re-run to enable SSH access. SMTP mail access is permitted to the designated server only. HTTP access is allowed on port 80 only.

This firewall script is very aggressive and generates lots of logs that are very detailed. This script is not very susceptible to getting out of order. Except for the last three lines, where packets are sent to the Firewall chain for denial.

Firewall rulesets:

1) IP Chains is used for all these examples
2) Kernel 2.2.15mdksecure is in use for the firewall.
3) Cisco router is used to provide first stage protection.

Firewall testing:
The best way to test your configuration is to run a port scan using nmap or another similar tool that will run several tests at the same time. Obviously this script for the firewall blocks most things, minimal testing is required for debugging. More testing is required for verifying that all the applications are working properly and aren't too restricted. Spoofing can be accomplished using packet forging programs.

Cisco Router Tips:
http://cio.cisco.com/warp/public/707/3.html

Chris Brenton's Networking haven ….
http://www.geek-speak.net

Linux Firewall Manager - Seawall:

http://seawall.sourceforge.net

Acknowledgements:
Various sites on the Internet were used to gather information for this document including but not limited to:

http://www.cisco.com various Cisco and Internetworking concepts
http://advice.netice.com great index of exploits, ports, preventions
http://www.ibm.com support indexes for SMB protocol orgination
http://support.microsoft.com listing of various services and protocols used for MS products

Brian M. Estep