Global Information Assurance Certification Paper

GIAC Firewall and Perimeter Protection Practical
By
Jay Swofford

Assignment 1: Egress Filter – 10 points

Egress filtering is the method by which a network administrator verifies that all packets exiting to the Internet have valid source addresses. The only reason not to implement egress filtering is that you have an overworked router that cannot handle the extra CPU cycles. However, if you are in this scenario you need to fix the router (upgrade, reconfigure, etc.) and then implement egress filtering. It should add no more than 1% maximum to your CPU utilization unless you are the source of a spoofing attack.

There are two excellent reasons to implement egress filtering. The first is to prevent your site from being the source of spoofing attacks on other sites (ranging from DDOS attack to your personal hacker/janitor late at night). The second is to prevent leakage of your internal addressing scheme to the Internet.

To implement this on a Cisco Router IOS version 12.0, you need to be in enable mode. You should already know how to get in enable mode, if you do not, you are not qualified to be configuring a live router. We will assume that your network is the 198.198.198.0/24 network address space.

```
Router#configure terminal
Router(config)#access-list 1 permit ip 198.198.198.0 0.0.0.255 any
Router(config)#access-list 1 deny any any log-input
```

The command is broken down as follows:

`Configure terminal` allows us to configure the router from the terminal session. This modifies the currently running configuration, so be careful!

`Access-list` is the actual Cisco command to add a line entry to an access-list

`1` is the number of the access list. 1-99 should be used since we are only sorting on IP address (standard access list).

`Permit/deny` tells the router what action to take when it receives a matching packet.

`198.198.198.0` is the network that you wish to screen on.

`0.0.0.255` is the mask used to verify a match. Note that this is the inverse of a normal network mask.

`Any` is a wild card key word that will match any IP address/Network mask combination (depending on the placement in the command line, it refers to first source and then destination address).

`Log-input` is a keyword that is appended to the normal command line to log all actions that match the given criteria.

Therefore, we have an access list that allows any source IP from our network to the Internet and that denies and logs all spoofed or invalid addresses. Next, we must apply this

access-list to an interface. The best place to apply it is on the interface that touches our internal network. If we place it on the interface touching the Internet, then we have wasted CPU cycles routing a packet destined to be dropped. We will assume that FastEthernet 0/0 faces your network.

```
Router(config)#interface fastEthernet 0/0
Router(config-if)#ip access-group 1 in
Router(config-if)#^Z
Router#write memory
```

Interface allows us to configure an interface on the router.

FastEthernet 0/0 is the name of the interface we wish to configure

Ip is the Cisco keyword to configure IP on the interface.

Access-group this parameter tells the interface that we wish to apply an access-list to the interface.

1 is the access-list number that we wish to apply. Note that this list must already exist for the command to succeed.

In tells the router whether the access-list should be compared to inbound or outbound traffic. In this case we are applying it to inbound traffic.

^Z (Control-Z) is the command to exit the configuration mode and place us back in enable mode.

Write memory is the command to write the new configuration to memory so that it will be there the next time you reboot.

To test this we can use two methods. The first is to disable NAT on one of our network segments and try to reach the Internet. The second is to use a packet generator (PacketX on NT) to spoof the source address of some other network. Since we are logging all denied packets we should immediately see entries in log. To do this, run the following command from enable mode:

```
Router#show access-lists 1
```

This will result in a printout to the terminal of access list 1 showing the number of times each rule was used. Hence, if it is working we should see an increase in the number of times that the 'deny any' rule (we are not logging the 'allow 198.198.198.0' rule) was used. The other way to check for success, is that you will not receive echo reply packets from the target network.

Assignment 2: Firewall Policy Violations – 50 points

These entries are all from the same firewall. The fields are Date, Time, Action, Source Address, Source Port, Source Network, Destination Address, Destination Port, Destination Network, and Rule Number. Rule 0 is Source * (any) to Destination LAN (internal network), drop the packet and log. Basically, this rule drops all traffic inbound from the Internet, unless a higher rule specifically allowed access. I am using a Webramp 700s from Ramp Networks. In all cases, I changed my source IP to a randomly selected IP of 198.198.198.1. Bad Guy networks have been translated to 66.66.66.66

Violation 1:

```
05/20/2000 04:08:41.016 -
      TCP connection dropped -
      Source: 66.66.66.66, 3305, WAN -
      Destination: 198.198.198.1, 98, LAN -     -     Rule 0
```

This is a very interesting entry. First, let's look at the source. According to http://whois.apnic.net (Asian-Pacific Network Information Center), this address is owned by a telecom company in Taiwan. This is even more interesting when you look at the destination port. 98/TCP (and 98/UDP) are well-known port numbers for TAC News. According to http://www.l0pht.com/pub/blackcrwl/telecom/arpa.txt:

> TACNEWS is a NIC (network information center) online service that offers login help to TAC users, includes the current list of ARPANET and MILNET TAC phone numbers, and provides a mechanism for reading the DDN (Defense Department Network) Newsletters and the DDN Management Bulletins. Users should read these publications regularly to stay current on DDN policies, announcements, and network news items. Access TACNEWS by logging into a TAC and typing "@n" or by using the TELNET service to connect to host SRI-NIC (10.0.0.51) and typing "tacnews".

No real risk here, since I am not running a DOD system. They were most likely scanning large subnet ranges for DOD systems.

Connecting to the source IP with telnet identifies it as running Red Hat Linux release 6.0 + CLE v0.8, Kernel 2.2.5-15CLE on an i686. Connecting to the source IP with FTP reveals that the hostname of the system is evilmachine.badguysrus.net and is running version wu-2.4.2-VR17(1). Connecting with Telnet to the SMTP service shows they are running Sendmail 8.9.3/8.9.3. Looks like a standard Linux box setup. However, it seems to be giving out a lot of information for a hacker's point of origin, so it might, and I want to reiterate 'might', be a compromised system.

Read any good spy novels lately?

P.S. I have a number of hits on my home system looking for port 98/TCP. All of them source from countries with known drug cartel links and/or histories of nefarious activities (Netherlands, Taiwan, Turkey, Mexico and Switzerland)

Violation 2:

```
05/19/2000 01:34:06.048 -
      TCP connection dropped -
      Source: 66.66.66.66, 3867, WAN -
      Destination: 198.198.198.1, 27374, LAN -    -     Rule 0
```

Now we are playing a little dirty, this one is a Sub7 v2.1 trojan attack. The source of this one is dialup.someisp.net.

This appears to be an attempt to contact a working version of the Sub7 server module (port 27374 is used by a Sub7 client to contact the server software). If the firewall had failed to stop this attack, it can be safely assumed that the perpetrator would have most likely attempted to install the trojan, after discovering that it did not already exist on my system. That would be Bad (notice the capital B). However, since my systems have properly layered defenses and all systems have current virus scanners with on-access

scanning, the overall risk is low. For more information on this trojan, see http://www.sans.org/y2k/subseven.htm. Currently there is no CVE designation for this trojan.

Violation 3:

```
05/18/2000 13:43:05.912 -
      UDP packet dropped -
      Source: 198.198.198.2, 2647, WAN -
      Destination: 198.198.198.1, 5632, LAN -    -    Rule 0
05/18/2000 13:43:39.672 -
      UDP packet dropped -
      Source: 198.198.198.2, 2648, WAN -
      Destination: 198.198.198.1, 5632, LAN -    -    Rule 0
05/18/2000 15:57:03.736 -
      UDP packet dropped -
      Source: 198.198.198.2, 2851, WAN -
      Destination: 198.198.198.1, 5632, LAN -    -    Rule 0
```

A little change of pace on this one. At first, this looks highly suspicious. Here is a user on my local subnet repeatedly hitting my port 5632. Port 5632? What's this? A quick search of the Internet (http://www.google.com) and we discover that this is probably an innocent "attack". Actually this (5632) is the port used by PCAnywhere32 7.52 or above to locate other PCAnywhere hosts on the local subnet. According to the Symantec customer support site, PCAnywhere, by default, sends out 254 UDP packets per network to form a browse list of available hosts. Setting HKLM\Software\Symantec\pcAnywhere \CurrentVersion\System\TCPIPNetBroadcast to 0 stops this behavior.

This could also be a hacker looking for non-password protected installations of PCAnywhere. If a hacker succeeds in connecting to my installation of PCAnywhere, then they have full remote control of my system. There is no CVE designation for this, since it is a feature of the software. However, there are three CVE candidates for this product: CAN-2000-0273, CAN-2000-0300, and CAN-2000-0324.

Violation 4:

```
05/21/2000 11:06:03.576 -
      TCP connection dropped -
      Source: 66.66.66.66, 65535, WAN -
      Destination: 198.198.198.1, 53, LAN -    -   Rule 0
```

This is an attempted TCP connection to my Domain Name Service (DNS). The extremely high source port number is suspicious and may mean this is a constructed packet. There are three scenarios for this packet.

1. This is a TCP DNS query. This is unlikely, since it should have been preceded by a failed UDP query. If this is all it is, then little to no risk to my home systems.

2. This is an attempted zone transfer of my registered DNS domain. Since my DNS zones are hosted by my ISP, this is of little risk. If I did have DNS and they did get a zone transfer, then they would have a nice network map of my site. Once again this scenario is unlikely due to the high source port number. This has a candidate CVE designation of CAN-1999-0532.

3. This is a constructed packet designed to cause damage (Denial of Service, reboot, etc.) to my non-existent DNS server. This would be my guess. Something similar to CAN-1999-0011.

The source address is cablemodem.somecity.se. This address is owned by a cable company in Sweden. This is the only hit I have seen from this range of addresses, so it was probably someone who found a cool DNS kill packet and was hitting US websites on port 53 to see if they could do anything.

Overall, this was a low risk, since none of my internal machines are listening on port 53.

Violation 5:

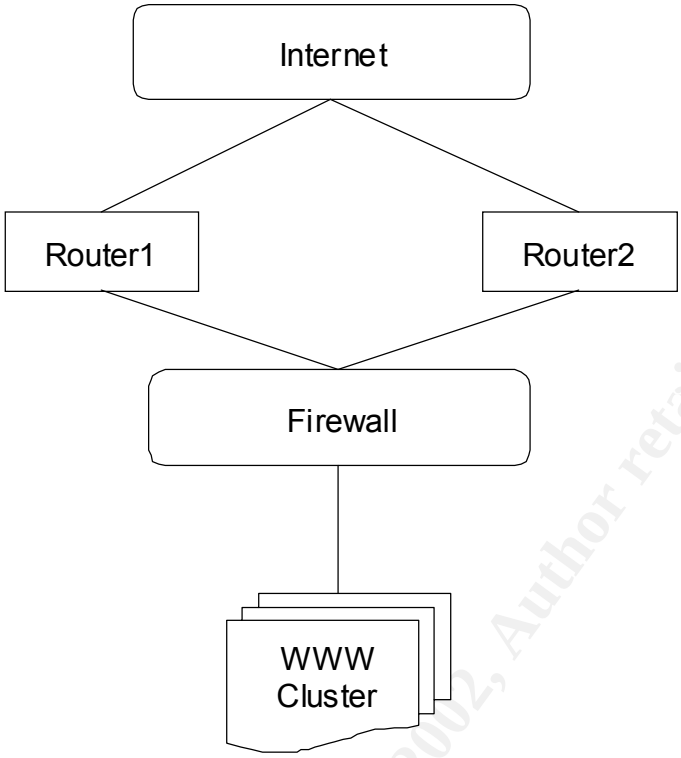
```
04/26/2000 02:31:49.912 -
      TCP connection dropped -
      Source: 66.66.66.66, 2666, WAN -
      Destination: 198.198.198.1, 111, LAN -     -     Rule 0
```

Here we have an attempted connection to port 111. According to the port number list at isi.edu, this is the Sun RPC (Remote Procedure Call) assigned port. The address is in Korea, so it may be related to the events captured by Jeff Stott and posted on the SANS web page on March 15, 2000 (there is an earlier posting on SANS for this kind of scan on January 21, 2000). Since I am not running any machines with this service, there would have been no effect if the firewall had not stopped the attack. If I did have this service running they (the bad guys) would have probably tried to exploit the service by making RPC calls to my boxes. Depending on which calls worked, they could have executed code, logged in, obtained interactive sessions, etc. All very bad. RPC is a service, not a vulnerability. Administrators who do not know how to correctly configure the service for safe use introduce the risk.

Assignment 3: Defense in Depth Architecture – 20 points

DDOS Resistant Architecture Design – 10 points:

There are two main concerns when defending against DDOS attacks. The first is spoofed address and the second is broadcast amplification. One lesser concern, but just as effective is bandwidth or CPU saturation. If you have older hardware or small pipes to the Internet, this may become your Achilles' heel. We will assume an internal address space of 198.198.198.0/24.



The two lines (each from a different ISP) connecting the Internet to the routers are DS3 (45 MB lines) choked down to only 6 MB/s each. Each line is subscribed to with the ISP as burstable lines. This means that when the traffic exceeds 6 MB/s on that link the line will continue to operate up to 45 MB/s. This means that a DDOS attack must sustain 90 MB/s to saturate your Internet connection.

On the ports connecting the Internet to the routers, we will apply ingress filtering on inbound traffic. Ingress filtering blocks all traffic that is not originating from valid Internet addresses. Based on the nice list at http://www.sans.org/dosstep/index.htm, our ingress filter would look like:

```
access-list 10 deny ip 192.168.0.0 0.0.255.255 any log-input
access-list 10 deny ip 172.16.0.0 0.15.255.255 any log-input
access-list 10 deny ip 10.0.0.0 0.255.255.255 any log-input
access-list 10 deny ip 0.0.0.0 0.255.255.255 any log-input
access-list 10 deny ip 127.0.0.0 0.255.255.255 any log-input
access-list 10 deny ip 198.198.198.0 0.255.255.255 any log-input
access-list 10 deny ip 169.254.0.0 0.0.255.255 any log-input
access-list 10 deny ip 192.0.2.0.0 0.0.0.255 any log-input
access-list 10 deny ip 224.0.0.0 15.255.255.255 any log-input
access-list 10 deny ip 240.0.0.0 7.255.255.255 any log-input
access-list 10 deny ip 248.0.0.0 7.255.255.255 any log-input
access-list 10 deny ip 255.255.255.255 255.255.255.255 any log-input
access-list 10 permit any any
```

This denies traffic originating from broadcast and private address spaces. It also denies traffic that is claiming to have originated from my own subnet. (To prevent DDOS as well as compromising UNIX trust relationships based on IP address). Based on real-world statistics, I have ordered the entries based on the number of hits they receive. This minimizes the CPU overhead generated by using this list. This list can be expanded to block the Internet Assigned Number Authority (IANA) reserved address spaces (Go to http://www.arin.net/whois/index.html and search on IANA).

We also want to disable directed broadcasts on the border router. To do this on a Cisco router we issue the following commands:

```
Router#configure terminal
Router(config)#interface ATM0
Router(config)#no ip directed broadcast
.
.      (repeat last two steps for each interface on your system)
.
Router(config)#^Z
Router#write memory
```

As recommended by SANS, the best method to test this is to ping the network based IP address of your site (198.198.198.0) and the broadcast address (198.198.198.255) of your network. The latter is best done from third party sites (e.g., http://nitrous.digex.net). But be careful, because if the ping succeeds you may end up on blacklists.

The next step is to apply egress filtering on the traffic entering the router from our DMZ. See question 1 for the specifics on this filter. Also see question 1 for tips on testing both of these filters. More refined alterations to Cisco router configurations can be found at http://www.cisco.com/warp/public/707/newsflash.html.

Next, we insert a proxy-based, redundant firewall into the path. The firewall blocks all traffic except for ports 80 (HTTP) and 443 (SSL) that is targeted to our web cluster address. This blocks all traffic originating from valid IP addresses, but is not intended for the services I provide. This also blocks all the known ports used to communicate between DDOS server and client installations. This helps prevent my network from contributing to an attack against myself or another domain.

Optionally, we can configure the firewall or insert an Intrusion Detection System (IDS) to watch for known DDOS attack signatures. Depending on the software used, we can configure it to dynamically change the access lists in the routers to block the source IP addresses for a period of time (defaults to 2 hours in the software I have used). This can be done with products like the Cisco PIX firewall or the ISS RealSecure IDS software. This allows us to kill the attacking zombies, without permanently blocking those addresses access to our web site. These should be configured to drop the packets (no returns since it just creates overhead for you). They should not be configured to do anything to the source IP addresses. Once you have initiated a packet back out to the Internet, you just crossed a very fuzzy legal line. If you made a mistake in the source addresses or beat up an innocent zombie then lawsuits and serving French fries may be in your future.

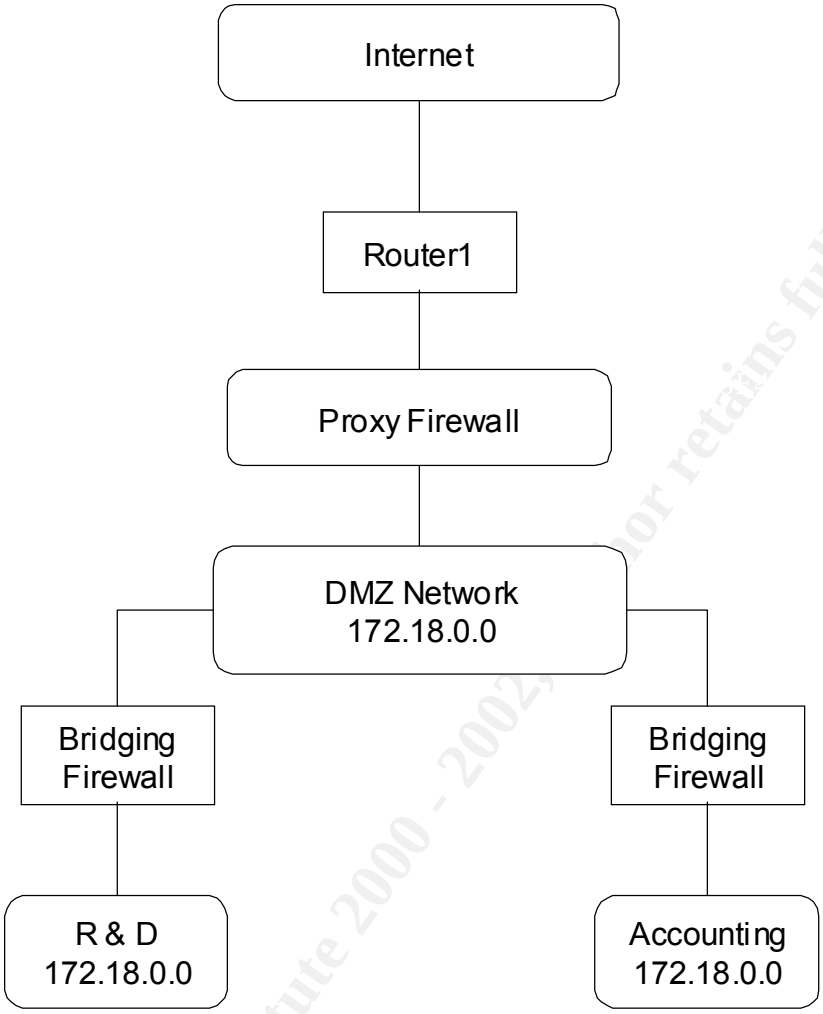
Lastly, we have a cluster of web servers connected to a load balancer (i.e., BigIP, Alteon, LocalDirector, etc.). This will effectively distribute the DDOS attack across many machines. This minimizes the impact on any one machine. It also allows us to rapidly add

machines to absorb the attack. Many of these systems can provide basic firewall, filtering and defensive actions as well as their load balancing capabilities.

Depending on the OS used, there are a number of configuration options available to decrease the time that ports are left in the open state. Some OS manufacturers have also released patches to harden the system against DDOS attacks. Make sure your OS and web software is up to date.

In summary, we used burstable bandwidth, ingress and egress filtering, firewalls, load balancing and OS configuration to dramatically decrease the effects of DDOS attacks.

High Security Architecture Design – 10 points:



This solution assumes a public address space of 198.198.198.0/24 and an internal, private address space of 172.18.0.0/16. To provide the greatest security for the two subnets we will need to layer our defenses and isolate the physical subnets.

Per the diagram, the router will act as a border router to the Internet. Ingress and egress filtering will be applied as in the last question. The external card will be the address assigned by our ISP. The internal NIC will have the address 198.198.198.1. This provides us with a border router that screens out bogus, spoofed and broadcast traffic.

Our proxy-based firewall will be the primary defensive layer against the Internet. Here would apply rules to drop all traffic except what is necessary. By default, we drop all inbound and outbound traffic. Then we open only those ports specifically necessary to conduct business with the Internet.

We would then place the two bridging firewalls on the physical wires leading to the Accounting and Research & Development subnets. This will be done so that the only physical path to these subnets goes through the bridging firewall. These two subnets should be in separate, physically secure areas within the office.

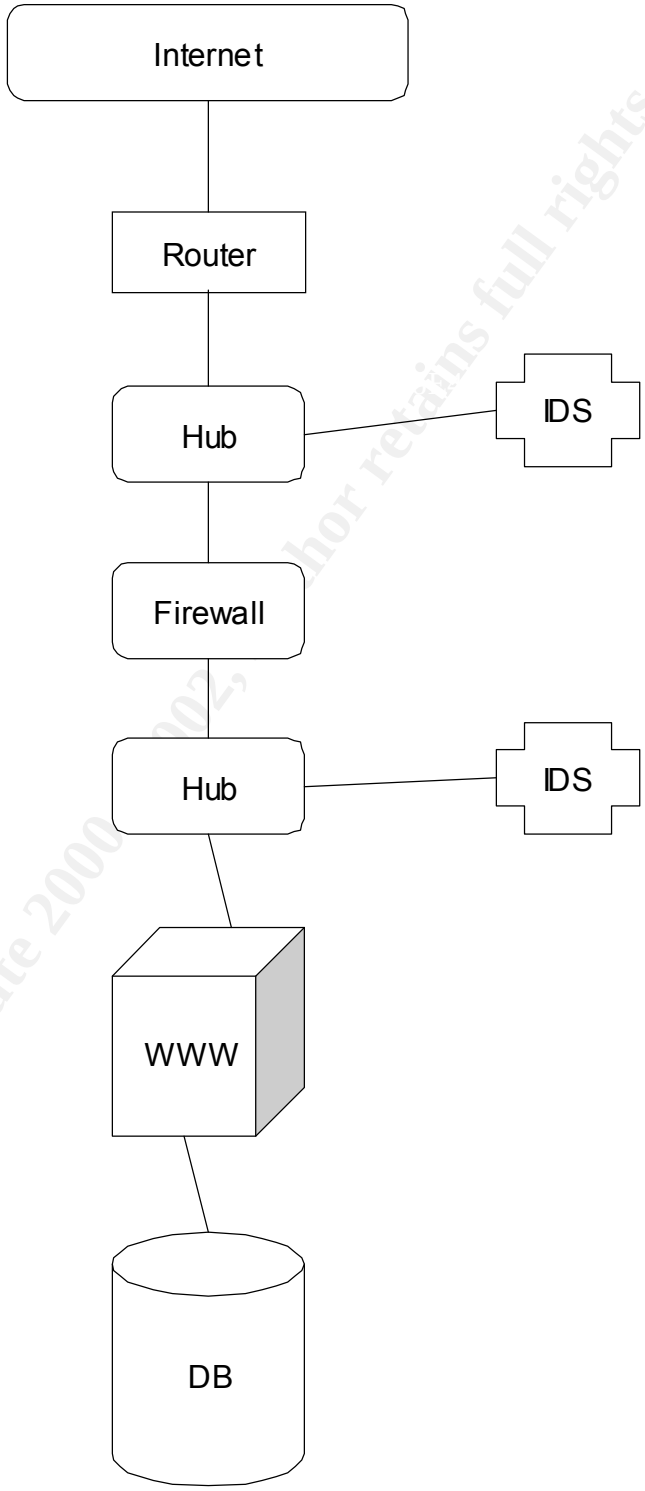
These firewalls would be the only real defense against internal threats. We would only allow outbound traffic to the main office as required to perform the job functions of the staff on those networks. Therefore, we choose to let the subnets to see out (to send mail only, if possible). All traffic destined for the net would be blocked.

Therefore, we have three layers of defense for each subnet. The first layer is the border router. The second layer is the proxy based firewall and the last layer is the bridging firewall providing local protection.

It should be noted that I assumed that all systems were properly configured for host based security and that proper internal security (password policies, logon hours, etc.) had been applied.

Assignment 4: Create a test that demonstrates your knowledge of the subject area – 20 points

Design a layered defensive structure for an e-commerce web site. You have a single connection to the Internet and want to provide maximum security for customer data. Assume that all hosts are properly configured and have all patches applied. Do not include redundancy for this question.



The first layer is the border router. All configuration is done through a local COM port. The border router is configured with ingress and egress filtering as in Assignment 3 part 1 and Assignment 1. This eliminates the majority of the noise originating from the Internet. However, we add to this configuration to block all IANA reserved addresses in the ingress filter. These addresses are (from http://www.arin.net/whois/index.html):

```
access-list 10 deny ip 192.0.0.0 0.0.255.255 any log-input
access-list 10 deny ip 128.9.64.26 255.255.255.255 any log-input
access-list 10 deny ip 191.255.0.0 0.0.255.255 any log-input
access-list 10 deny ip 223.255.255.0 0.0.0.255 any log-input
access-list 10 deny ip 128.0.0.0 0.255.255.255 any log-input
access-list 10 deny ip 1.0.0.0 0.255.255.255 any log-input
access-list 10 deny ip 0.0.0.0 0.255.255.255 any log-input
access-list 10 deny ip 96.0.0.0 31.255.255.255 any log-input
access-list 10 deny ip 65.0.0.0 31.255.255.255 any log-input
access-list 10 deny ip 23.0.0.0 0.255.255.255 any log-input
access-list 10 deny ip 31.0.0.0 0.255.255.255 any log-input
access-list 10 deny ip 197.0.0.0 0.255.255.255 any log-input
access-list 10 deny ip 201.0.0.0 0.255.255.255 any log-input
```

The border router would connect to a hub or VLAN (switched environment). Connected to this hub would be our first intrusion detection workstation. This workstation would be plugged into the hub with a NIC configured to be in promiscuous mode. The workstation would have a second NIC that connects to the same network as our management station. This allows us to see what kinds of attacks are being launched against our site from the raw Internet connection. It is configured to alert when it identifies Denial of Service attacks or attacks aimed at firewall software.

Also plugged into this hub is our firewall. We have chosen routing firewall. Due to the high traffic overhead of e-commerce traffic, proxy-based firewalls have shown themselves to be a bottleneck (based on my own testing, I noticed up to a 10% delay in page paints during peak traffic loads). This firewall will perform 1:1 NAT for all machines on the DMZ (In this example, this is just the load balancer, but in real life we would also have DNS and Mail on the DMZ). The only physical path from the Internet to the DMZ is through the firewall. Here are ruleset is:

Rule 2: Source Any; Destination Web; Port 80, 443; Allow

Rule 1: Source Any; Destination Firewall; ICMP Request, ICMP Reply, Allow

Rule 0: Deny All

All firewall management will be handled from the local interface, so we do not need rules to allow this. Rule 2 allows all traffic on 80 and 443 to our web site. Rule 1 allows customers to ping the NAT IP addresses of our web site. Rule 0 dumps all other traffic. We do not have a business need to see outside. Therefore, we have not included rules to allow for this.

Directly connected to the internal interface of the firewall is another hub or VLAN (switched environment). Connected to this hub is our second Intrusion Detection workstation. This one is configured the same as the first. Except this one is set to send alerts on any attack signature that may cause damage to our site. For instance, if I am running a pure NT environment, I do not need to send alerts for Sun RPC attacks, but I do need to send alarms for WinNuke attacks.

Also connected to this hub or VLAN is our load balancer. This machine has a private address (remember the 1:1 NAT on the firewall?). Here we again provide NAT services. Our load balancer translates its external private address to the private address of our multiple web servers. The web servers use a different private address space, to further befuddle and confuse attackers ("Slow'em down with inane paranoia" is my motto). Our load balancer is configured to only allow ports 80 and 443 to pass. All configuration is done through the local COM port.

Our web servers are dual-homed (two NICs). The second NIC is connected to a hub or VLAN that has another, unique private address space. On this network resides the database servers that contain customer information.

With properly configured users, passwords and systems this provides very good security for an e-commerce web site. By creating unique layers with security between each layer, we have developed a defensible web site.

There are only two paths to the customer data at this point. The easiest one is social engineering to gain physical access to the database servers (e.g., phone technician is here to inspect the telecom switch in your server room). The only other path is through the layered defense structure. Like any security, it still can be done, but with all those layers, the intrusion detection systems or log auditing should spot any intruder before they discover and compromise all layers. The ingress filtering at the border router has forced the attacker to use a legitimate public IP address so we should have no trouble tracking them back to the launch point of the attack.