Global Information Assurance Certification Paper

GIAC Firewall and Perimeter Protection Curriculum

Practical Assignment for SNAP San Jose May 8-13, 2000

Prepared by Layne Bro
06/14/2000

Assignment 1 – Egress Filter

Egress Filtering is the practice of filtering outbound traffic from your network. This traffic may be headed to the Internet or to another internal network. Most filtering is only done on inbound connections since most security setups are only worried about incoming traffic. The point of egress filtering is to control traffic headed out of your network. Through the use of Egress filtering, an organization can be more aware of malicious internal users and/or potentially compromised systems. Egress filtering can be beneficial both to the organization performing the filtering and to the Internet community at large.

An attackers number one priority is to avoid detection and being caught. Therefore, if the attacker can use your network to hide his identity, he'll be happy to do so. Without egress filtering, an attacker is free to use your network to launch spoofing attacks against other systems and sites. With egress filtering, spoofed packets are not allowed out and a log entry can be made by the filter to alert you to a malicious internal user or possibly a compromised system within your organization.

Syntax of the filter

For the following discussion, lets assume that the legal address space for your organization is 100.100.100.0. On a Cisco router (the last hop before leaving your organization), you can create an extended access list with the following command (in configuration mode):

Router (config) # access-list 112 permit ip 100.100.100.0 0.0.0.255 any

Basically, the access-list command creates a new access-list referenced as 112. This access list will permit all ip traffic with a source ip address of 100.100.100.x and destined for anywhere.

Now that we have a valid access list, we need to apply it to an interface. We would apply this to the internal interface (lets say eth1) by typing the following commands (again, in configuration mode):

int eth1
ip access-group 112 in

The int eth1 command tells the router that you want to deal with the eth1 interface only. The ip access-group 112 in command says that you want to apply access-list 112 to all inbound packets on this interface (eth1).

This has now enabled the access list on the internal interface of the router on all inbound traffic. In other words, as soon as the router accepts the packet, it will compare it with this access list. If the packet does not match this access list, it will be dropped without any further processing required.

Now that we have a valid access list and have applied it, lets take a look at a command to log all violations of this policy. You can create a log entry for every violation of the above policy by adding the following line to your access list (again, in configuration mode):

Router (config) # access-list 112 deny ip any any log

This list simply explicitly states that all traffic not handled by a previous rule should be denied and a log entry created. In a Cisco router, the final rule is an implicit deny, but by using the explicit statement of this rule combined with the log parameter, you can now log all traffic attempting to violate this rule. With this log file, you should be able to (at your leisure ☺,) track down the source of the traffic and determine if you have a compromised system or a malicious internal user attempting to attack another site.

Let's put all the pieces together. Here are the commands to create an egress filter that logs all violations (in command mode):

Access-list 112 permit ip 100.100.100.0 0.0.0.255 any

```
Access-list 112 deny ip any any log
Int eth1
Ip access-group 112 in
Exit
Exit
Show running
```

This set of commands will create the access list 112 and enter your two rules, permit valid outbound traffic and log all other traffic. It will then apply this access list to the eth1 interface on inbound traffic. After applying the list, you then exit out of configuration mode for interface eth1 and then out of configuration mode. Then, by entering show running, you will be able to see the current configuration of your Cisco router.

Testing your filter

After creating and enabling your egress filter, you'll want to verify it is working properly. To do this, you'll first want to verify that your valid traffic is making it to its destination. Take one of your internal boxes and make sure it can still reach destinations on the Internet. After making sure you're not stopping valid traffic, you should make sure you are stopping the traffic you want to and that you are logging it if applicable.

Probably the easiest, most low cost way to test this setup is to try and send reserved addresses out to the Internet. If you are running your internal network with any of the reserved address ranges, you can disable NAT and see if the router drops and logs your packets. If you are not running NAT, you could also try setting up a box in your DMZ with an invalid IP address and try sending packets from it. A final test should be to use a packet spoofer and try to send data through your router. Not only should it drop the packets, it should also be logging every attempt. If it isn't, you need to go back and find out where the configuration is incorrect and start over.

Jun 6 23:58:37 194.247.87.235:53 -> z.y.w.34:53 FIN ***F****

The section "Jun 6 23:58:37" is the timestamp
The section "192.247.87.235:53" is the source IP address and Port
The section "z.y.w.34:53" is the destination IP address and Port
The section "FIN ***F****" is the description of the attack

This attack would be picked up by a rule watching for FIN packets for a nonexistent connection. In other words, you could be using a stateful inspection firewall, which would notice that this FIN packet to close the session is actually not related to an existing connection.

If this attack had been successful, the destination system would have responded with a RESET/ACK packet and the attacker would now know that port 53 is listening on this system. This would most likely imply that DNS is running on this server and is worthy of more attention to gain more information about the victim network. Also, now that the attacker knows that port 53 is listening on this system, they can use other tools to attack this particular port and service.

Jun 7 17:15:29 212.160.91.50:30864 -> z.y.w.34:53 SYN **S*****

"Jun 7 17:15:29" is the timestamp
"212.160.91.50:30864" is the source address and port
"x.y.z.53" is the destination address and port
"SYN **S*****" is the description of the attack with the bits shown

This attack would be picked up by a rule watching for TCP connections to the DNS service.

If this attack had succeeded, the attacker could have possibly done a zone transfer of the entire DNS contents. This is a common method of reconnaissance. The more an attacker can learn about your network, the better prepared he is to attack. A zone transfer would have told him all of the systems on your network (that DNS knew about), what their names were and their IP addresses.

Jun 7 17:15:27 212.160.91.50:53 -> z.y.w.98:53 UDP

"Jun 7 17:15:27" is the timestamp
"212.160.91.50:53" is the source IP Address and Port
"z.y.w.98:53" is the destination IP address and port
"UDP" is the packet type

A filter denying all inbound UDP traffic would pick up this attack.

This packet registers as a simple DNS query. In this instance, lets assume that this box is not providing the DNS services for outside users. Therefore, the packet is likely a request to see if there is a system listening on this port. If there is a system but its not listening on this port, the attacker will receive a return packet telling him that there is a valid system, but its not listening on this port. If there is no system at this address, the attacker will receive a response telling him the host doesn't exist. In either case, he has learned something new. If the host doesn't exist, he doesn't need to spend any more time trying to attack this address. If the host does exit but isn't running a listener on port 53, the attacker now knows that there

is a valid host to be attacked but that DNS tools won't work. Either way, the attacker has gained info about the victim network.

Jun 6 22:59:26 213.6.15.254:58110 -> z.y.w.98:21 SYN 2*S***** RESERVEDBITS

"Jun 6 22:59:26" is the timestamp
"213.6.15.254:58110" is the source IP Address and Port
"z.y.w.98:21" is the destination IP address and port
"SYN 2*S****RESERVEDBITS" is the attack description along with the bit settings

A filter looking for invalid bit settings in the TCP header would pick up this attack.

This attack is an attempt to send bogus TCP flags to the destination and receive some type of data in reply. Sometimes this type of attack will result in a response that can be linked to a specific OS and the attacker can then use this information to decide which tools and exploits to try next.

Jun 6 22:59:26 213.6.15.254:58111 -> z.y.w.98:21 NULL ********

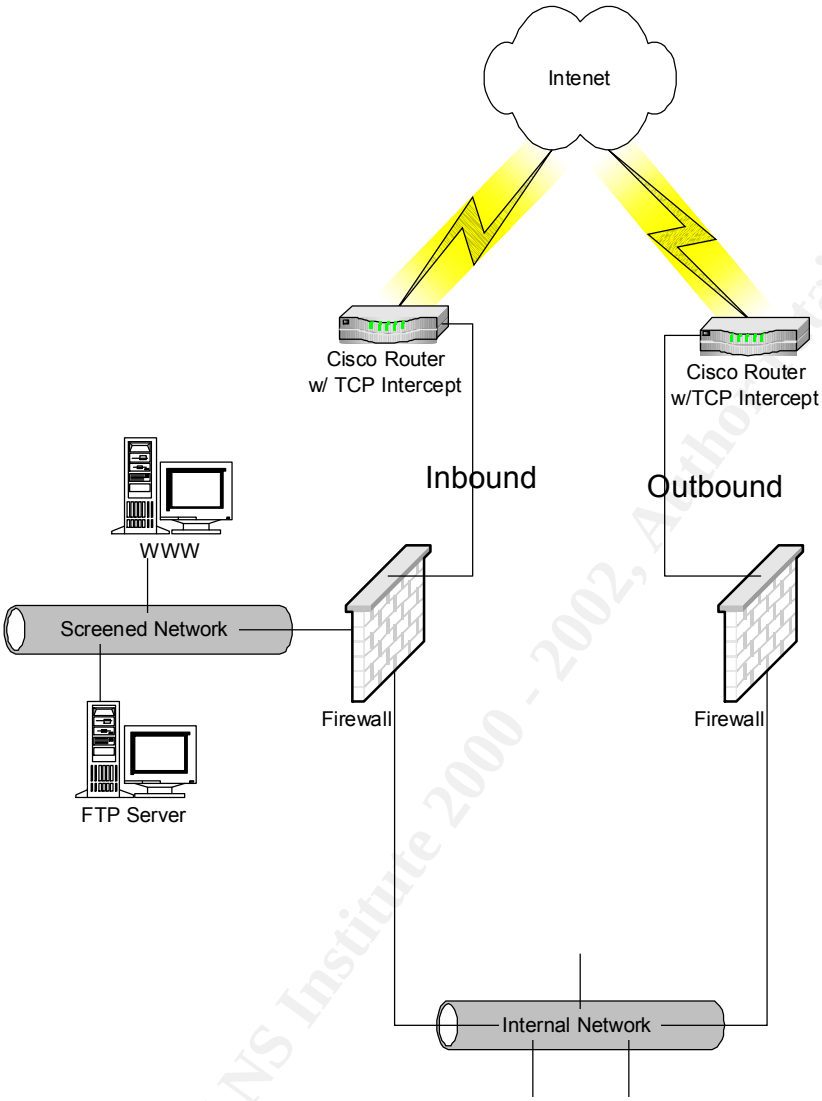
"Jun 6 22:59:26" is the timestamp
"213.6.15.254:58110" is the source IP Address and Port
"z.y.w.98:21" is the destination IP address and port
"NULL ********" is the attack description along with the bit settings

A filter looking for invalid bit settings in the TCP header would pick up this attack.

This is also an attack to try and determine something unique about the system based on the response to bogus bit flag settings. If the attacker can get the victim host to reply to this packet with any kind of information, he may be able to determine the OS or other characteristics about the host. He can then use this information to aid in his overall attack by using specific tools and exploits that this system is potentially vulnerable to.

Part I

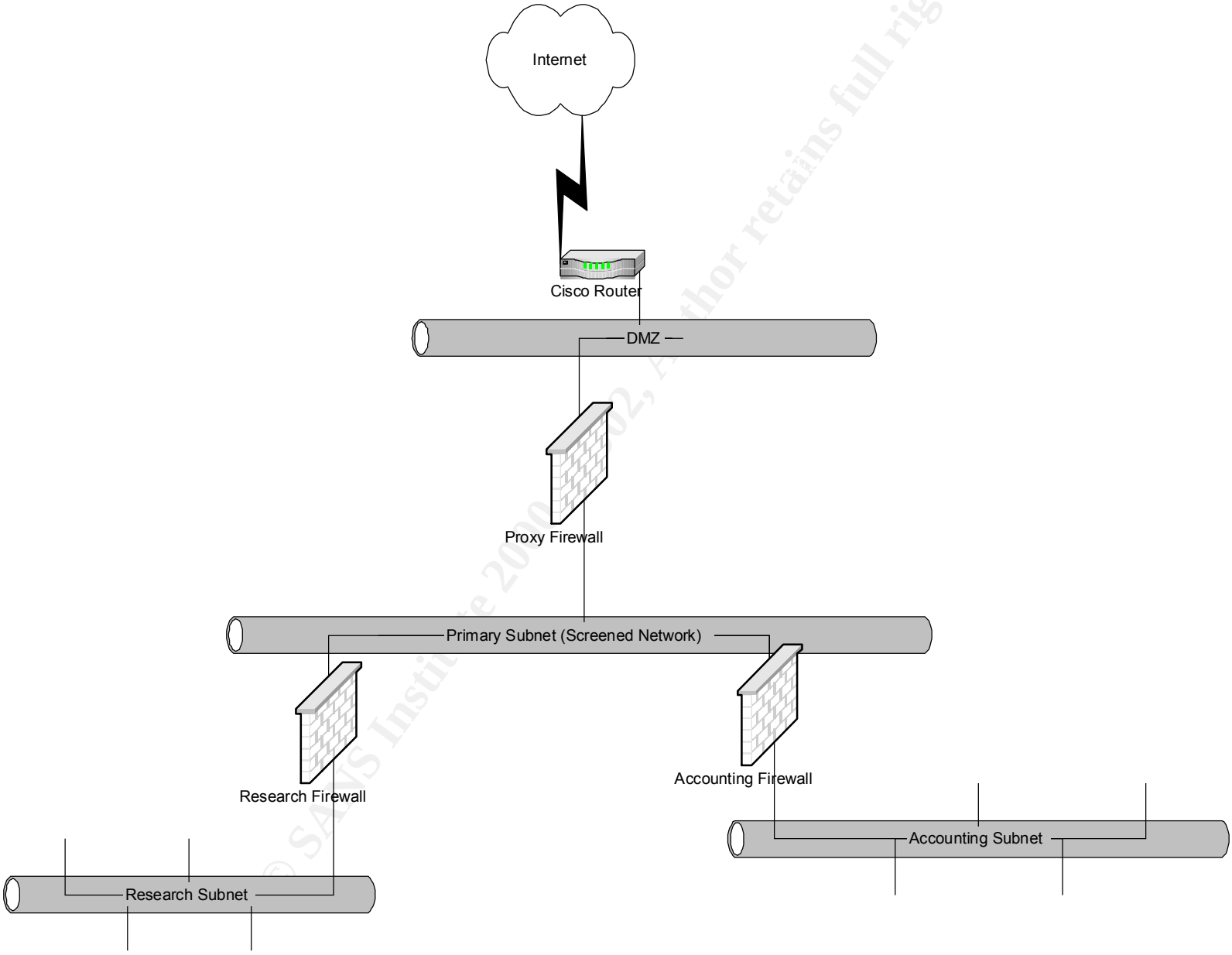
There is a site with dual Internet connections and we want to design a system to be resistant to DDOS Attacks.



I would recommend setting up the two Internet connections to perform different functions. One line could be used for all outbound traffic from the internal network and the other could be used for all inbound traffic to the domain name. On each line I would recommend setting up the Border Gateway Router with the normal configuration (i.e. drop all reserved addresses, perform egress filtering, drop all broadcast traffic) and using a Cisco Router. On both routers I would also enable TCP intercept. TCP Intercept is specifically used to prevent Denial of Service attacks. Behind each router, I would recommend a firewall along with a screened network for services supplied to the Internet (www, email, etc.).

Part II

We have a site with two critical subnets that need protection. A previous employee had already ordered a Cisco Router, one Proxy Firewall, and two appliance type firewalls with 2 10/100 NICs capable of performing in a bridging manner. This equipment is already onsite and needs to be used in an effective design for protection.



First, we will want to setup the Cisco Router as our Border Gateway router. We will connect the serial interface to the line from our ISP and the Ethernet interface to our DMZ. The DMZ will contain our valid address space. On the Border Gateway router, we will want to perform Egress filtering as well as denying all reserved address spaces (RFC1918).

Behind the Border Gateway Router, we will setup our Proxy firewall. We will use this box to control all access into and out of our internal subnets. This box will perform Network Address Translation by default since it is a proxy type firewall. Proxy firewalls have an "Air Gap" in that no data actually passes through the box. Rather, the Proxy server receives a request for data, goes out and retrieves the data itself and then sends the data back to the original requestor. This is actually more secure than a packet filtering type firewall in that no data actually crosses between subnets. There is a need to make sure the box running the proxy server has enough resources to not be the bottleneck.

Inside of the proxy firewall, we will have our first internal subnet. Hanging off of this internal subnet will the firewall appliances for both the Research and the Accounting subnets. These appliance type firewalls will help separate traffic. Traffic only intended for the accounting department will not leak out to the primary subnet or cross into the research subnet. The Research subnet will have similar levels of protection. These appliance type firewalls will also provide a second layer of security for an attacker to penetrate. They are probably also a different technology than the proxy firewall and will require an attacker to spend more time and knowledge to penetrate the research and/or accounting subnets. In addition, the appliance type firewalls could also be setup to perform NAT (Network Address Translation) thereby limiting the amount of information an attacker could gather about the research and accounting subnets. By performing NAT, the primary subnet doesn't even know what the IP address ranges of accounting and research are. An attacker couldn't sit inside the primary subnet and try to sniff the packets for this information. Also, by having the multiple layers of firewalls, internal users are also prevented from "sniffing" information from the research or accounting subnets.

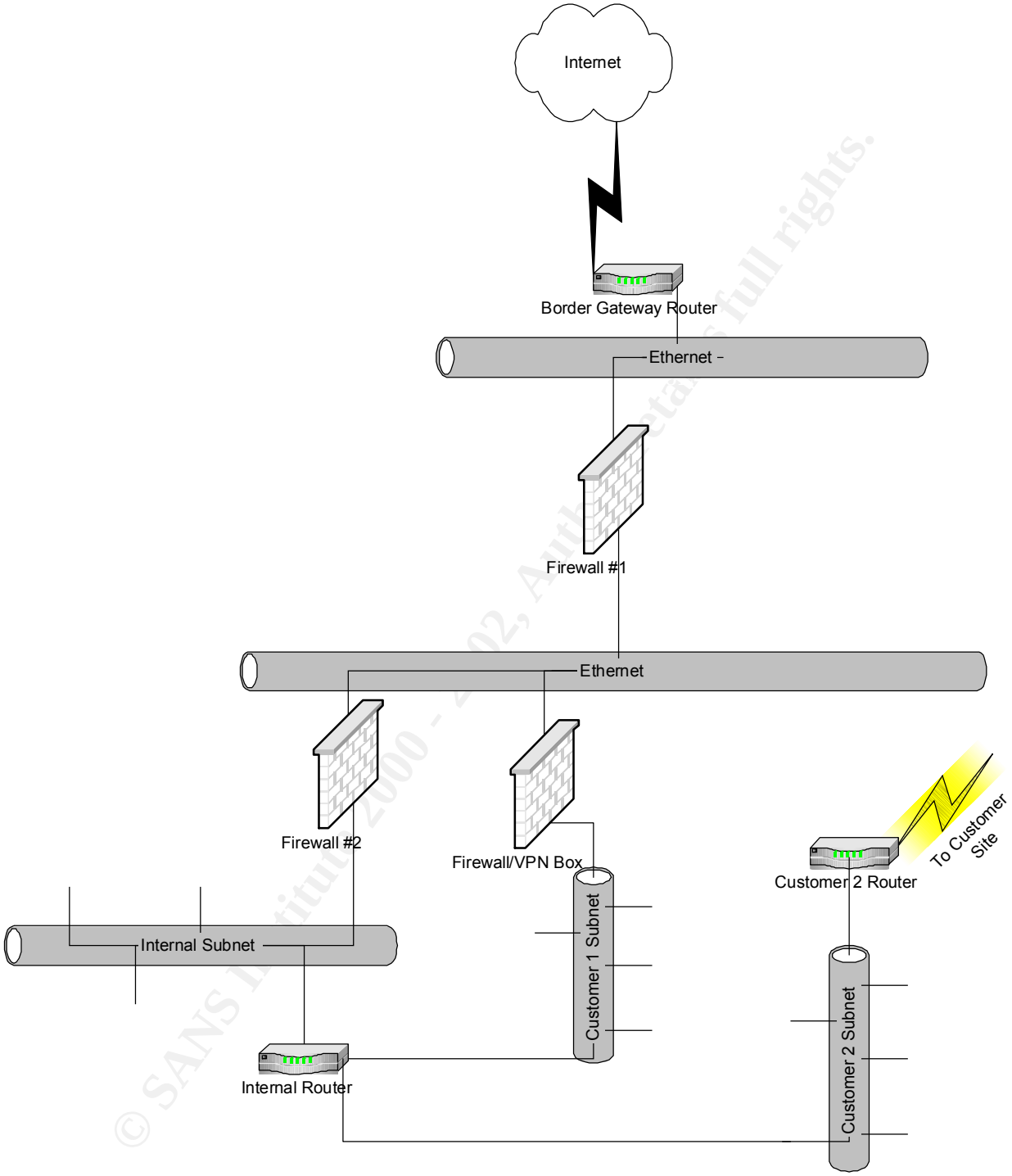
Finally, the three layers of security (Border Gateway Router, Proxy Firewall, Appliance type Firewall) should all be logging information. By reviewing these logs independently and as a cohesive unit, the site administrator should be able to detect an intruder long before they are able to penetrate the research or accounting networks.

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

Submit a detailed design for the following scenario:

You work for a development shop where you develop ecommerce sites for multiple clients on their hardware but at your office. You also need access out to the Internet for your developers but, since this is a development shop and not a production environment, you don't need the Internet to access the sites you are building. You do need to allow access from your customers to view your work. For financial reasons, one of your customers wants to use a VPN connection and the other has opted for a dedicated circuit. You have one office of internal developers along with 2 customers. How do you design a network for maximum flexibility while still providing maximum protection for your own systems and for your customers systems on your network?

Interesting dilemma. Seems that we need to purchase 3 Cisco Routers, 2 firewalls, and a Firewall/VPN box. Using this setup, we should be able to provide maximum flexibility while still providing security from the Internet and between projects.



On the Border Gateway Router, we would need to setup Egress Filtering, destination broadcast addresses filtering, and also deny all reserved address spaces (RFC 1918).

On Firewall #1, we would need to setup rules that only allowed inbound access to the firewall/vpn box. We should also setup a reflexive rule that verified any return traffic to firewall #2 was actually return traffic. Our third rule should be to drop and log any other traffic trying to pass through. On Firewall 2, we

would need to enable NAT to verify that only our valid address space was trying to make it to the Internet. In addition, we could setup Egress filtering if we had a problem on the Border Gateway with an internal user trying to spoof traffic. We should create a reflexive rule to allow outbound traffic to any Internet site with any protocol. Finally, we should create a deny all rule and log all packets that this rule drops.

On the firewall/VPN box, we would of course need to setup the VPN software. In addition, we should setup rules to drop all other traffic and log it. There is no need for the customer subnet to access the Internet nor for the Internet to directly access the customer subnet, so we should drop and log any traffic trying to violate the policy. We would NOT need to setup NAT on this box since no traffic should actually be passing through. The VPN traffic will be automatically translated to the local valid Internet address due to the nature of VPNs

On the internal Router, we would need to setup ACLs to indicate that only traffic from the internal subnet to the customers' subnet is allowed. We could setup a reflexive ACL to allow outbound traffic with responses properly returned to the sender on the internal subnet. We would need to drop and log any inbound traffic from a customer segment to the internal subnet. Also, we should drop all broadcast packets and perform egress filtering to verify we don't have an internal user trying to attack customers. It would also be wise to drop all source-routed packets, as this could also be used to attack customers. A possible addition to the internal router would be NAT. If we NAT'd all internal subnet addresses before they were sent to the customer subnets, the customer subnets and routers wouldn't need to know our internal addressing scheme.

This router should only know about the local devices on the customers' subnet and the route back to the customer site. This way, a malicious user at the customer site couldn't easily sniff packets to learn more about our internal subnet. We would want to provide egress filtering and logging on this router. In addition, we could enable NAT, but it might be helpful to know where the connections to the customer servers are coming from (assuming the Site admin at the customer is not performing NAT on the way out of his network). Any packets specifically aimed at our internal router originating at the customer site should also be dropped.