Global Information Assurance Certification Paper

GIAC Level Two Firewalls, Perimeter Protection, and VPN's

Practical Assignment for Capital SANS
Dec 10-15, 2000

Matthew E. Leiby

Submitted 2/18/01

Assignment

1

Security Architecture

Define a security architecture for GIAC Enterprises, a new Internet startup that expects to earn $200 million per year in online sales of fortune cookie sayings.

Produce a diagram or set of diagrams with explanatory text that define how to use perimeter technologies to implement your security architecture.

You must consider and define access for:

Customers (the companies that purchase bulk online fortunes);

Suppliers (the authors of fortune cookie sayings that connect to supply fortunes);

Partners (the international partners that translate and resell fortunes).

Your architecture must specify filtering routers, firewalls, VPNs to partners, secure remote access, and internal firewalls. Be explicit about the brand and version of each perimeter defense component.

My network diagram is located on the last page of this practical. Please make reference to it when needed.

All of the domain names, host names and IP Addresses used in this assignment are fictitious. Any similarities to existing companies are purely coincidental. As of Jan 1st, 2001, 4chewns.com has not been registered.

Being a new startup company, GIAC Enterprises/4chewns.com will be using VPN technology to leverage the Internet for secure and cost effective connectivity to International Partners and Suppliers. Since our Customer base is International we will be supporting Standard Internet Browsers with the maximum allowed International encryption level to browse and place orders online to our SSL Secured Web Server.

The same can be said for International Suppliers and Partners with regard to the VPN encryption level that we use

The ISP that GIAC Enterprises has chosen has supplied a 32-address subnet for our use on our Internet Presence. Private address space will be used for all internal devices, with one exception, which will be explained later.

Assume that the domain 4chewns.com was registered and guessmy.4chewns.com is listed as the Primary DNS and one provided by the ISP as the Secondary DNS. Assume that we have acquired a Digital Certificate from a prominent Certificate Authority to support SSL for our Web Server.

We will be using a Cisco 3620 router with IOS version 12.04(T) for the Border Router and a Cisco PIX Firewall with version 5.3(1). The Cisco 3620 router is running a version of the IOS that supports the Cisco IOS Firewall Feature Set. The Cisco PIX will also be used for the VPN Gateway supporting Telecommuters (Client to Gateway), and our Partners and Suppliers (Gateway to Gateway).

All PC's are loaded with PC Firewall Software for defense in depth, and Virus Detection Software, that is being centrally managed. In the case of Telecommuters, the PC Firewall Software is the first line of defense. All Telecommuters are required to use a Strong Authentication Token (SecureID) to authenticate against the Cisco PIX during IPSEC negotiations. I will not be showing the tacacs+ server or SecurID server on the Diagram since it was getting pretty cluttered. Rest assured that they are on the Internal network. We haven't bought into the idea of using Digital Certificates for our VPN Telecommuters or VPN Gateways yet and will be using preshared keys for the IKE negotiation.

We have chosen not to use a reverse proxy firewall that would provide an additional layer of protection for access to our Web Server located in Service Net 1. Instead, we have posted our fairly static product line on a web server provided by our ISP. When potential customers decide to order, they are directed to the SSL secured web server in Service Net 1. It is my understanding that a reverse proxy server would not provide any additional protection that is not provide by the Cisco Pix for SSL traffic since the traffic it would be handling is encrypted.

IP address assignment

The IP address space that is provided by our ISP is XXX.240.1.0/27, a 32 address subnet.

We will further subnet this for use in Service Net 1, Service Net 2, Service Net 3, The Cisco Pix Firewall VPN tunnel address, and it's NAT and PAT addresses.

XXX.240.1.0/29 – Service Net 1

- XXX.240.1.1 – www.4chewns.com
- XXX.240.1.2 – mailrelay.4chewns.com
- XXX.240.1.3 – dns2.4chewns.com
- XXX.240.1.4 – Unassigned
- XXX.240.1.5 – Service Net 1 PAT Address
- XXX.240.1.6 – Firewall Service Net 1 Interface

XXX.240.1.8/29 – Service Net 2

XXX.240.1.9 – DB.4chewns.com, WebServer Database Server

XXX.240.1.10 – Unassigned

XXX.240.1.11 – Unassigned

XXX.240.1.12 – Unassigned

XXX.240.1.13 – Unassigned

XXX.240.1.14 - Firewall Service Net 2 Interface.

XXX.240.1.16/30 – Service Net 3 (And the Internal DNS Subnet!)

XXX.240.1.17- guessmy.4chewns.com, Internet DNS. (And Internal DNS!)

XXX.240.1.18 – Firewalled Ethernet Interface 0/1 on the Border Router. (And InternalRouter1 Ethernet Interface)

XXX.240.1.20/29 - Firewall NAT and PAT

XXX.240.1.21 NAT Address for Internal Syslog Server

XXX.240.1.22 NAT Address for Internal NTP Server

XXX.240.1.23 NAT Address for Internal TFTP Server

XXX.240.1.24 dev.4chewns.com, Development Server

XXX.240.1.25 PAT Address assigned to all internal devices accessing the Internet that are not defined for Static NAT.

XXX.240.1.26 Border Router Ethernet Interface 0/0 – Secondary IP address.

XXX.240.1.28/30 – Firewall Outside Interface (VPN Tunnel)

XXX.240.1.29 Firewall Outside Interface.

XXX.240.1.30 Border Router Ethernet Interface – Primary IP address.

YYY.1.1.0/24 – Internal Network addresses

I will show only one internal router which is just to demonstrate the correlation between the YYY.1.1.0/24 network and the Internal DNS subnet of XXX.240.1.16/30.

Don't be alarmed that we seem to have two subnets on the diagram using the same address space. Let me point out that we are NAT'ing all traffic between the internal network and the Internet. There is no possible way for my Internal network to mistakenly talk to the Internet DNS, nor for my Internal DNS to not get to the Internet properly. This is because the Internal DNS is forwarding all requests not in it's domain to the recursive DNS that sits in Service Net 1, which makes the requests on it's behalf. If it were used for anything else to get to the Internet it would pick up a translation provided by PAT (Port Address Translation).

Note - There is a current Cisco PIX bug that requires a static NAT statement for any Internal DNS's that NAT going through the PIX. This has to do with the inability to use udp port 53 properly with DNS responses when using PAT (Port Address Translation. We will not be changing the IP addresses for traffic between the Internal Network and Service Net 1 or 2. There are static NAT rules that we will be defining in order to do this that will be described later.

We have several reasons for creating our DNS structure like this. The primary reason being early VPN Client implementations.

1. Some VPN clients do not support assigning a DNS during IPSEC negotiation. By having our users hardcode X.240.1.17 as their Primary DNS, they can use their PC at work, properly resolving Host names to IP addresses. When they happen to be out of the office, using any ISP access method, they still have a valid DNS to make requests against. With their VPN deactivated, they can resolve all Internet visible hostnames, including our own, against the External DNS in Service Net 3. When the VPN is activated, the tunnel is created to the Cisco PIX, and the Cisco PIX only knows to route to the XXX.240.1.16/30 subnet by sending it to InternalRouter1, therefore it will resolve against the Internal DNS. The latest CiscoSecure VPN3000 Client actually downloads the DNS and WINS information during IPSEC negotiation, which makes this point no longer valid. It sure helped prior to this version.

2. Connections to partners can sometimes get a little funny. Let's say that we decide to create a VPN to a partner and that partner demands that we not NAT our IP addresses. Not only that, we have to provide valid DNS resolution for all hosts or servers that may connect to their network. Don't laugh, I had this happen.

3. Mergers and Acquisitions can be a tough time for us IT folks. We decide to merge with Company B. Upper management wants the network to look seamless. Again, with our Internal DNS being the same one that we have registered as our Internet visible DNS, we do not need to immediately merge the DNS structure. It can take a long time to weed out duplicate host names and such.

** Please note that with scenario 2 and 3, that the partners Internet connection must be working in order for their DNS to query the Root Servers to get the correct DNS to resolve your host names to addresses. This is not a perfect solution, but I thought I would share it. Maybe it will give you some ideas of your own.

Assignment

2

Security Policy

Based on the security architecture that you defined in Assignment 1, provide a security policy for AT LEAST the following three components:

Border Router

Primary Firewall

VPN

You may also wish to include one or more internal firewalls used to implement defense in depth or to separate business functions.

By 'security policy' we mean the specific ACLs, firewall ruleset, IPSec policy, etc. (as appropriate) for the specific component used in your architecture. For each component, be sure to consider internal business operations, customers, suppliers and partners. Keep in mind you are an E-Business with customers, suppliers, and partners - you MAY NOT simply block everything!

(Special note VPNs: since IPSec VPNs are still a bit flaky when it comes to implementation, that component will be graded more loosely than the border router and primary firewall. However, be sure to define whether split-horizon is implemented, key exchange parameters, the choice of AH or ESP and why. PPP-based VPNs are also fully acceptable as long as they are well defined.)

For each security policy, write a tutorial on how to implement each ACL, rule, or policy measure on your specific component. Please use screen shots, network traffic traces, firewall log information, and/or URLs to find further information as appropriate. Be certain to include the following:

1. The service or protocol addressed by the ACL or rule, and the reason these services might be considered a vulnerability.

2. Any relevant information about the behavior of the service or protocol on the network.

3. The syntax of the ACL, filter, rule, etc.

4. A description of each of the parts of the filter.

5. An explanation of how to apply the filter.

6. If the filter is order-dependent, list any rules that should precede and/or follow this filter, and why this order is important. (Note: instead of explaining order dependencies for each individual rule, you may wish to create a separate section of your practical that describes the order in which ALL of the rules should be applied, and why.)

7. Explain how to test the ACL/filter/rule.

Be certain to point out any tips, tricks, or "gotchas".

Border Router

I have attempted to use the SANS Institutes Top Ten Blocking Recommendations as a baseline for my Ingress and Egress Access Lists. I have a few exclusions and will explain them as appropriate. I decided to paste the configuration here and try to explain the commands line by line since some of them don't really pertain to just Filtering or Firewalling. I used the Command Reference Master Index located at http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/12cgcr/rbkixol.htm to help me with my Border Router configuration and my explanations. I also make reference the Cisco PIX Command Reference, which is located at: http://www.cisco.com/univercd/cc/td/doc/product/iaabu/pix/pix_v53/config/commands.htm

service timestamps debug datetime msec localtime show-timezone
service timestamps log datetime msec localtime show-timezone

I configured timestamping on this router to log messages in localtime. NTP is configured to keep the router in time. The NTP configuration is shown later in the configuration. SANS Top Ten #9 says I should turn NTP off, however, I find it useful for my needs. I will be blocking other devices from using me as a timeserver, as well as configuring a trusted NTP source.

Syntax: service timestamps type datetime [msec] [localtime] [show-timezone]

Syntax Description

Type	Type of message to timestamp: debug or log.
Uptime	(Optional) Timestamp with time since the system was rebooted.
Datetime	Timestamp with the date and time.
Msec	(Optional) Include milliseconds in the date and timestamp.
Localtime	(Optional) Timestamp relative to the local time zone.
show-timezone	(Optional) Include the time zone name in the timestamp.

service password-encryption

I configured password encryption for this router to keep unauthorized individuals from viewing the passwords in the local configuration or by viewing the configurations that are saved to my configuration backup server (tftp server). Cisco adds this warning to the Command Reference:

"Caution

This command does not provide a high level of network security. If you use this command, you should also take additional network security measures."

In other words, it keeps honest people honest.

hostname borderrouter

I named my router "borderrouter". The host name is used in prompts and default configuration filenames.

logging buffered 4096 informational

I limited messages logged to the routers internal buffer based on the severity "informational" and held it at a the maximum buffer size of 4096 bytes. I use this for a quick, near real-time look at what the router is doing. I change this level based on what I want the router to record locally at that time.

Valid levels are:

Level Arguments	Level	Description	Syslog Definition
Emergencies	0	System	LOG_EMERG
Alerts	1	Immediate action needed	LOG_ALERT
Critical	2	Critical conditions	LOG_CRIT
Errors	3	Error conditions	LOG_ERR
Warnings	4	Warning conditions	LOG_WARNING
Notifications	5	Normal but significant	LOG_NOTICE
Informational	6	Informational messages	LOG_INFO
debugging	7	Debugging messages	LOG_DEBUG

enable secret 5 xxxxx (encrypted password x'ed out)

Cisco has you create two passwords when initially configuring a router. I chose not to add the complexity of privilege levels since only a few individuals have the ability to work on this router, and they need to have full Exec-Mode privileges. In short, privilege levels allow you to assign defined levels of abilities on the router per the password that is used.

Per the Cisco Documentation:

"The enable secret command provides better security by storing the enable secret password using a non-reversible cryptographic function. The added layer of security encryption provides is useful in environments where the password crosses the network or is stored on a TFTP server."

Additional Cisco Documentation:
"If you use the same password for the enable password and enable secret commands, you receive an error message warning that this practice is not recommended, but the password will be accepted. By using the same password, however, you undermine the additional security the enable secret command provides.
Note - After you set a password using enable secret command, a password set using the enable password command works only if the enable secret is disabled or an older version of Cisco IOS software is being used, such as when running an older rxboot image. Additionally, you cannot recover a lost password that has been encrypted by any method."

In other words, it is possible for the encrypted enable password to be reversed into plain text, and if that matches your enable secret, then you really don't have a non-reversible enable secret password. Don't do it!
Syntax: enable secret [level level] {password | [encryption-type] encrypted-password}

enable password 7 xxxxxx (encrypted password x'ed out)

Once again, I chose not to use privilege levels. The enable password is stored and displayed in encrypted form because of the earlier set "service password-encryption" command. This password will display in clear text if the "no service password-encryption" command is used. Privilege level 15 is the default (traditional enable privileges). In this case, the 7 refers to the encryption type. You will not ordinarily enter an encryption type. Typically, you enter an encryption type only if you copy and paste into this command a password that has already been encrypted by a Cisco router.
A Caution from the Cisco Documentation:
"Caution
If you specify an encryption type and then enter a clear text password, you will not be able to reenter enable mode. You cannot recover a lost password that has been encrypted by any method."
Syntax: enable password [level level] {password | [encryption-type] encrypted-password}

clock timezone EST –5

I set the time zone for display and logging purposes.
Syntax: clock timezone zone hours [minutes]

zone	Name of the time zone to be displayed when standard time is in effect.
hours	Hours offset from UTC.
Minutes	(Optional) Minutes offset from UTC.

clock summer-time EST recurring

I also set the time for automatic daylight savings time. I specified "clock summer-time zone recurring" without additional parameters to default to the summer time rules for the United States. The actual syntax will allow you to specify other dates and time that you want daylight savings time to turn on and off. Either of these may be used.
Syntax:
clock summer-time zone recurring [week day month hh:mm week day month hh:mm [offset]]
clock summer-time zone date date month year hh:mm date month year hh:mm [offset]
clock summer-time zone date month date year hh:mm month date year hh:mm [offset]

no ip source-route

This is part of #1 in the SANS Top Ten Blocking Recommendations Using Cisco ACL's. It discards any IP datagrams that contain a source-route option.

no ip finger

To deny Finger protocol requests to be made of the router, I used the "no service finger" global configuration command. If enabled, this service is equivalent to issuing a remote "show users" command. It would display a list of logged on users and their IP addresses or Hostnames.

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

SANS Institute - Top Ten Blocking Recommendations Using Cisco ACL's refers to adding the "no service tcp-small-servers" and the "no service udp-small-servers" commands. The Cisco documentation states that regardless of whether you have the no option defined or not, the current settings will be displayed. Not in my version! The "no" options are not being displayed. I will be checking the Cisco Bug Tracker very soon!
These commands disable the TCP and UDP servers for Echo, Discard, Chargen, and Daytime services.
When these services are disabled, access to Echo, Discard, and Chargen ports cause the Cisco IOS software to send an "ICMP port unreachable" message to the sender and discard the original incoming packet. If they were being displayed properly, they would have looked like this in my configuration.

ip domain-name 4chewns.com

The ip domain-name command make's it easy to resolve any Internet visible hostname within my domain of 4chewns.com.
Syntax is simply: ip domain-name *name*
Where *name* is the Default domain name used to complete unqualified host names.

ip name-server XXX.240.1.17

The ip name-server is used to let me resolve my companies Internet visible hostnames and any other Internet host against the guessmy.4chewns.com DNS. guessmy.4chewns.com is locally attached to interface E/0/1, which is labeled as ServiceNet1 on the Diagram.
Syntax: ip name-server *server-address1 [server-address2...server-address6*] Additional
Server addresses can be added for Secondary use in case of a failure of the Primary. Each address will create an additional configuration line entry for each Server. The Name Servers will be queried in the order that they are entered.

ip inspect audit-trail

The next few "ip inspect" commands exist only to support my Internet visible DNS on Service Net 3, which is located on ethernet interface 0/1. You must use an IOS version with the IOS Firewall Feature Set in it which is capable of performing these Firewall functions. The technology used is described as "Context Based Access Control", or CBAC. The Cisco Documentation can describe it better than I can.
"CBAC intelligently filters TCP and UDP packets based on application-layer protocol session information and can be used for intranets, extranets and internets. Without CBAC, traffic filtering is limited to access list implementations that examine packets at the network layer, or at most, the transport layer. CBAC inspects traffic that travels through the firewall to discover and manage state information for TCP and UDP sessions. This state information is used to create temporary openings in the firewall's access lists to allow return traffic and additional data connections for permissible sessions (sessions that originated from within the protected internal network)."
This should not be confused with an "established connection" rule. When using CBAC, the return rule is dynamically created based on a permitted packet that the router saw go past the interface.

With that said, here is how I implemented it.

"ip inspect audit-trail" turns on CBAC audit trail messages, which will be displayed on the console and log buffer after each CBAC session closes. The command is entered as shown with no arguments or keywords.
Here's what an audit trail message looks like in the log buffer and the syslog server:
Jan 1 11:00:58.755 EST: %FW-6-SESS_AUDIT_TRAIL: udp session initiator (bbb.bbb.bbb.bbb:1038) sent 31 bytes – responder (XXX.240.1.17:53) sent 163 bytes
I can only guess what's actually being resolved in the DNS query session (notice that this is a UDP session to my DNS's port 53). I would like to think that this is a potential customer querying my DNS for my web servers IP address so they can buy some fortune cookie sayings!

ip inspect name e01out udp
ip inspect name e01out tcp

I configured TCP and UDP inspection rules to watch TCP and UDP packets that leave interface ethernet 0/1 destined for my DNS on Service Net 3. When this inspect rule is actually applied to the interface, it will allow udp and tcp port 53 requests to respond back to permitted initiating devices. The permitted protocol and ports are defined in access list 101 which is applied inbound to interface ethernet 0/2. The part of access list 101 that defines this is shown here:
access-list 101 permit udp any host xxx.240.1.17 eq domain (anyone can do DNS lookups)
access-list 101 permit tcp host ddd.ddd.ddd.ddd host xxx.240.1.17 eq domain (ISP secondary DNS can do zone transfers)
The effect of this is that as the packet is watched leaving interface ethernet 0/1, CBAC will create a dynamic rule in access list 103 allowing the return traffic that is normally denied. The underlined part of the display below are the rules that were created dynamically by CBAC. The rest is the actual configured access list. Notice how CBAC places it's rules at the top of the access list. If you have ever configured access lists on Cisco routers, you know that order does matter. If CBAC were to place it's rules at the bottom, the packets would still be denied by the "deny ip host XXX.240.1.17 any" statement at the end.

```
borderrouter>show ip access-list 103
Extended IP access list 103
    permit udp host XXX.240.1.17eq domain host zzz.229.73.16 eq 4562 (1 match)
    permit udp host XXX.240.1.17eq domain host yyy.186.46.110 eq 1593 (1 match)
    permit udp host XXX.240.1.17any eq domain (3619 matches)
    permit tcp host XXX.240.1.17eq telnet host XXX.240.1.18 (298 matches)
    permit udp host XXX.240.1.17eq domain host XXX.240.1.18 (6 matches)
    deny ip host XXX.240.1.17any log (9 matches)
```

Syntax: ip inspect name inspection-name protocol [timeout seconds]

ip inspect name e01in udp

I configured a UDP inspection rule to watch UDP packets that will enter interface ethernet 0/1 from my Internet DNS on Service Net 3. When this inspect rule is actually applied to the interface, it will allow UDP port 53 requests to respond back to my Internet DNS from the queried Internet device. The permitted protocol and port is defined in access list 103 which is applied inbound to interface ethernet 0/1. Here's the part of access list 103 that allows the DNS to make DNS queries to devices on the Internet:

access-list 103 permit udp host XXX.240.1.17 any eq domain

What happens here is access list 103 allows DNS query traffic in interface ethernet 0/1, CBAC then creates a dynamic rule to the top of access-list 101, which is applied as an inbound rule on interface ethernet 0/2. Here's and example of part of what access list 101 looks like when my Internet DNS makes a DNS query:

borderrouter>show ip access-list 101

Extended IP access list 101

permit udp host rrr.41.3.101 eq domain host XXX.240.1.17eq domain (3 matches)

permit udp host ttt.206.240.5 eq domain host XXX.240.1.17eq domain (1 match)

The rest of the static rules defined for access list 101 would start here, but is not displayed for brevity. Access List 101 is my Ingress filter.

interface Ethernet0/0

Describes the physical port on the router.

Description PIX Firewall subnet

Description simply gives the interface a name that others and I can readily understand.

ip address XXX.240.1.26 255.255.255.248 secondary
ip address XXX.240.1.30 255.255.255.252

I use two ip addresses on my interface that connects to my PIX firewall for one reason. So I can traffic shape (prioritize) traffic with some rules based on IP address. I will be using the XXX.240.1.28/30 network for my PIX firewalls Outside Interface IP address, which is used for the VPN tunneling. The XXX.240.1.20/29 network is for my PIX firewalls Static NAT addresses, and the PAT address. This configuration will keep my outbound surfers from using up all my bandwidth between my ISP and my border router. I will show the traffic shaping rules that are used on my end of the circuit later.

no ip redirects

no ip redirects is another recommended service to turn off as listed in the SANS Institute Top Ten Blocking Recommendations Using Cisco ACL's. If enabled, it would allow the sending of ICMP Redirect messages if the Cisco IOS software was forced to resend a packet through the same interface on which it was received.

no ip unreachables

no ip unreachables is another recommended service to turn off as listed in the SANS Institute Top Ten Blocking Recommendations Using Cisco ACL's. We do not want to send ICMP host or Protocol Unreachable messages. This is how the router would react to each if we left it turned on:

"If the Cisco IOS software receives a nonbroadcast packet destined for itself that uses a protocol it does not recognize, it sends an ICMP Protocol Unreachable message to the source.

If the software receives a datagram that it cannot deliver to its ultimate destination because it knows of no route to the destination address, it replies to the originator of that datagram with an ICMP Host Unreachable message."

no ip directed-broadcast

no ip directed-broadcast is another recommended service to turn off as listed in the SANS Institute Top Ten Blocking Recommendations Using Cisco ACL's. This command prevents any broadcast traffic from being forwarded. If turned on it would forward broadcast traffic for several services, like netbios, tftp, bootp, just name a few. In this scenario, we do not need to forward any of these broadcast packets.

no cdp enable

no cdp enable is another recommended service to turn off as listed in the SANS Institute Top Ten Blocking Recommendations Using Cisco ACL's. This disables the Cisco Discovery Protocol (CDP).

interface Ethernet0/1
description Internet Visible DNS
ip address 10.240.1.18 255.255.255.252

I think we get the idea on these commands now.

ip access-group 103 in

This applies access list 103 to inbound traffic on interface ethernet 0/1. This access list prevents the DNS server from initiating any traffic to an external device other than DNS queries on udp port 53. The additional rules allow an administrator on the border router to resolve hostnames to IP addresses, and to telnet to the DNS for administration.
access-list 103 permit udp host X.240.1.17 any eq domain (permits DNS queries to remote DNS's)
access-list 103 permit tcp host X.240.1.17 eq telnet host x.240.1.18 any (Permits the DNS to respond to telnet requests from the border router)
access-list 103 permit udp host x.240.1.17 eq 53 host x.240.1.18 any (Permits the DNS to respond to DNS queries from the border router)
access-list 103 deny ip host x.240.1.17 any log (logs all denied attempts)
Notice that there is no rule in here that states that the DNS can respond to queries to it's UDP port 53. This is dynamically added by CBAC when it sees valid traffic going outbound through the interface to the DNS. This was discussed earlier when describing the "ip inspect name e01out udp and ip inspect name e01out tcp" , global rules.

no ip redirects
no ip unreachables
no ip directed-broadcast

We've already discussed these statements.

ip inspect e01in in
ip inspect e01out out

This is where apply the ip inspect rules to the interface, and in which direction. In this case I have the ip inspect rule named e01in applied to inbound traffic on this interface. I also have an ip inspect rule named e01out applied to outbound traffic. If you remember from earlier the ip inspect rules looked like this:
ip inspect name e01out udp
ip inspect name e01out tcp
ip inspect name e01in udp

no cdp enable

We've already discussed this statement.

interface Ethernet0/2
description To ISP

This is my interface connected to my ISP. They gave me a Full Duplex 10baseT ethernet connection and I only bought part of that bandwidth, which is being Traffic Shaped. I will discuss how this is done a little further down the page.

ip address lll.186.6.2 255.255.255.252

I'm showing this IP address because our ISP wants to be able to ping this address to monitor the connection. You will see this address later in the Ingress filter.

ip access-group 101 in

Here's where I apply my Ingress filter. I will explain this better when I get to the access list 101 configuration. This is also where CBAC will dynamically add a return statement for DNS queries that originate at my Internet visible DNS.

ip access-group 102 out

Here's where I apply my Egress filter. I will explain this better when I get to the access list 102 configuration.

no ip directed-broadcast

We've already discussed this statement.

full-duplex

This circuit is running in full duplex mode. Be careful, not all interfaces are capable of full-duplex.

traffic-shape group 109 1000000 12000 12000 1000
traffic-shape group 110 1000000 36000 36000 1000

This is where we get what we pay for, and do a little traffic shaping to give us some designated bandwidth for certain IP addresses.
From the Cisco Documentation:
"Traffic shaping uses queues to limit surges that can congest a network. Data is buffered and then sent into the network in regulated amounts to ensure that traffic will fit within the promised traffic envelope for the particular connection.

Use traffic shaping if you have a network with differing access rates or if you are offering a subrate service. You can configure the values according to your contract with your service provider or the service levels you intend to maintain"
In my case, we bought 2MB worth of bandwidth and we want to split that up into two pieces. 1MB for our VPN access and access to our Service Net 1 servers. And another 1MB for our outbound Internet surfing use and Internet DNS. Which IP addresses use which chunk is defined in access lists 109 and 110.
access-list 109 permit ip XXX.240.1.0 0.0.0.7 any (Service Net 1 Subnet)
access-list 109 permit ip XXX.240.1.28 0.0.0.3 any (PIX Outside Interface Subnet which includes our VPN tunnel IP Address)
access-list 110 permit ip XXX.240.1.20 0.0.0.7 any (PIX NAT and PAT Subnet)
access-list 110 permit ip XXX.240.1.16 0.0.0.3 any (Service Net 3, Internet DNS Subnet)
Syntax: traffic-shape group access-list bit-rate [burst-size [excess-burst-size]]

access-list Number of the access list that controls the packets that traffic shaping is applied to on the interface.

bit-rate Bit rate that traffic is shaped to in bits per second. This is the access bit rate that you contract with your service provider, or the service levels you intend to maintain.

burst-size (Optional) Sustained number of bits that can be transmitted per interval. On Frame Relay interfaces, this is the committed burst size contracted with your service provider.

excess-burst-size (Optional) Maximum number of bits that can exceed the burst size in the first interval in a congestion event. On Frame Relay interfaces, this is the excess burst size contracted with your service provider. The default is equal to the burst-size.

My configuration has thrown some extra stuff (1000) at the end of each line, but I can't find out what it means. Another question for Cisco.
My ISP needs a rule very similar to this coming to me. The difference would be that the access lists would be in reverse order.
Example:
access-list 109 permit ip XXX.240.1.0 0.0.0.7 any
access-list 109 permit ip XXX.240.1.28 0.0.0.3 any
would look like this:
access-list 109 permit ip any XXX.240.1.0 0.0.0.7
access-list 109 permit ip any XXX.240.1.28 0.0.0.3
I have not included ServiceNet2 in any of these rules since there is no requirement for any devices in that subnet to ever talk directly to the Internet. If they did, the PIX rules would NAT them.

interface Ethernet0/3

This interface is unused. I could use this for testing VPN gateway to gateway connections. As you can see, I have the standard stuff turned off all the time.

no ip address
no ip redirects
no ip unreachables
no ip directed-broadcast
shutdown
no cdp enable

ip route 0.0.0.0 0.0.0.0 lll.186.6.1

This command tells my border router to forward all packets that it doesn't have a known route for, to route to my next hop at the ISP.

ip route XXX.240.1.0 255.255.255.248 XXX.240.1.29

This route statement tells packets destined for my Service Net 1, that the next router is my PIX Outside Interface.

no ip http server

This disables the ability to use an http browser to connect to the router for management. It is off by default.

logging history informational

This limits the levels of information sent to the syslog server. The levels were described earlier when discussing the "logging buffered 4096 informational" command

logging source-interface Ethernet0/1

All syslog messages will appear on the syslog server as coming from the primary IP address assigned to Interface ethernet 0/1.

logging XXX.240.1.21

This is the NAT IP address assigned on the Cisco PIX that correlates to the real IP address of my internal server that I use for syslogging.

access-list 1 permit XXX.240.1.21

This access list is applied to vty ports 0 thru 4, which only allows the NAT IP address of my syslog server to telnet to the Border Router. All others would be denied access. Here's a typical message that you would get from a Unix Server that is not included in this access list:
serverx% telnet X.240.1.30
Trying X.240.1.30...
telnet: Unable to connect to remote host: Connection refused

Preface: Access List 101 is applied as an inbound filter on interface ethernet 0/2 that is connected to my ISP and the rest of the world. This is my Border Routers Ingress filter. I have chosen to create entries for the SANS Top Ten Blocking Recommendations Using Cisco ACL's and am logging several of them. By listing them separately, I will be able to stop logging some of the more annoying and repetitious logging messages in the future and keep ones that I may deem as curious activity. This would be impossible if I simply made rules for only the traffic that I wanted to permit, then specified a "deny ip any any log" at the end

access-list 101 deny ip XXX.240.1.0 0.0.0.31 any log

SANS Top Ten Blocking Recommendations Using Cisco ACL's, #1
My publicly assigned IP Address Space.

access-list 101 deny ip 127.0.0.0 0.0.0.1 any log

SANS Top Ten Blocking Recommendations Using Cisco ACL's, #1
Common Server Loopback Addresses

access-list 101 deny ip 10.0.0.0 0.255.255.255 any log
access-list 101 deny ip 172.16.0.0 0.0.255.255 any log
access-list 101 deny ip 192.168.0.0 0.0.255.255 any log

SANS Top Ten Blocking Recommendations Using Cisco ACL's, #1
Private Address Space per RFC1918

access-list 101 deny ip 224.0.0.0 31.255.255.255 any log

SANS Top Ten Blocking Recommendations Using Cisco ACL's, #1
IANA reserved addresses

access-list 101 permit tcp any host XXX.240.1.1 eq 443

Permit ssl access to my web server on Service Net 1

access-list 101 permit tcp any host XXX.240.1.2 eq 25

Permit smtp access to mail relay server on Service Net 1

access-list 101 permit tcp any eq 25 host XXX.240.1.2 gt 1024

Permits the mail relay server responses due to the server sending email. Permits ephemeral ports only.

access-list 101 permit tcp any eq 53 host XXX.240.1.3

Allows TCP DNS responses back to DNS on Service Net 1

access-list 101 permit udp any eq 53 host XXX.240.1.3

Allows UDP DNS responses back to DNS on Service Net 1

access-list 101 permit tcp any host XXX.240.1.25 gt 1024
access-list 101 permit udp any host XXX.240.1.25 gt 1024

Permits replies to all ephemeral ports (ports greater than 1024) used by the Cisco PIX for the PAT IP Address. This rule must be added before any deny rules that would affect these ports that are applied later.

access-list 101 permit udp any host XXX.240.1.17 eq domain

This rule permits any host to do a DNS query against my Internet Visible DNS on Service Net 3. This rule must be added before a deny rule that would affect UDP port 53 for all addresses. CBAC will take care of creating a dynamic entry to access list 103 for replies to these connection requests.

access-list 101 deny tcp any any range ftp telnet log (21 thru 23)
access-list 101 deny tcp any any eq 139 log
access-list 101 deny tcp any any range exec cmd log (512 thru 514

SANS Top Ten Blocking Recommendations Using Cisco ACL's, #2

access-list 101 deny tcp any any eq sunrpc log (111)
access-list 101 deny udp any any eq sunrpc log (111)
access-list 101 deny tcp any any eq 2049 log
access-list 101 deny udp any any eq 2049 log
access-list 101 deny tcp any any eq 4045 log
access-list 101 deny udp any any eq 4045 log

SANS Top Ten Blocking Recommendations Using Cisco ACL's, #3

access-list 101 deny tcp any any eq 135 log
access-list 101 deny udp any any eq 135 log
access-list 101 deny udp any any eq netbios-ns log (137)
access-list 101 deny udp any any eq netbios-dgm log (138)
access-list 101 deny tcp any any eq 139 log
access-list 101 deny tcp any any eq 445 log
access-list 101 deny udp any any eq 445 log

SANS Top Ten Blocking Recommendations Using Cisco ACL's, #4

access-list 101 deny tcp any any range 6000 6255 log

SANS Top Ten Blocking Recommendations Using Cisco ACL's, #5

access-list 101 deny tcp any any eq domain log (53)
access-list 101 deny udp any any eq domain log (53)
access-list 101 deny tcp any any eq 389 log
access-list 101 deny udp any any eq 389 log

SANS Top Ten Blocking Recommendations Using Cisco ACL's, #6

access-list 101 deny tcp any any eq smtp log (25)
access-list 101 deny tcp any any eq pop2 log (109)
access-list 101 deny tcp any any eq pop3 log (110)
access-list 101 deny tcp any any eq 143 log

SANS Top Ten Blocking Recommendations Using Cisco ACL's, #7

access-list 101 deny tcp any any eq www log (80)
access-list 101 deny tcp any any eq 443 log
access-list 101 deny tcp any any eq 8000 log
access-list 101 deny tcp any any eq 8080 log
access-list 101 deny tcp any any eq 8888 log

SANS Top Ten Blocking Recommendations Using Cisco ACL's, #8

access-list 101 deny tcp any any range 1 ftp-data log (1 thru 20)
access-list 101 deny udp any any range 1 20 log (1 thru 20)
access-list 101 deny tcp any any eq 37 log
access-list 101 deny udp any any eq 37 log

SANS Top Ten Blocking Recommendations Using Cisco ACL's, #9

access-list 101 deny udp any any eq tftp log (69)
access-list 101 deny tcp any any eq finger log (79)
access-list 101 deny tcp any any eq nntp log (119)
access-list 101 deny tcp any any eq 123 log
access-list 101 deny tcp any any eq lpd log (515)
access-list 101 deny udp any any eq syslog log (514)
access-list 101 deny tcp any any eq 161 log
access-list 101 deny udp any any eq snmp log (161)
access-list 101 deny tcp any any eq 162 log
access-list 101 deny udp any any eq snmptrap log (162)
access-list 101 deny tcp any any eq bgp log (179)
access-list 101 deny tcp any any eq 1080 log

access-list 101 permit icmp host lll.186.2.129 host lll.186.1.2 echo log
access-list 101 permit icmp any XXX.240.1.0 0.0.0.31 packet-too-big log
access-list 101 permit icmp any host XXX.240.1.29 echo-reply log
access-list 101 permit icmp any host XXX.240.1.29 host-unreachable log
access-list 101 deny icmp any any

SANS Top Ten Blocking Recommendations Using Cisco ACL's, #11

My ISP wants to be able to ping my ethernet 0/2 interface, so I allow that from their Network Management Station. I also allow icmp packet-too-big responses to any of my addresses. I like to test ping from my firewall, so I am allowing echo replies back into my network to my Cisco Pix Outside Interface. The Cisco PIX Command Reference also recommends that you keep ICMP unreachable active.

> "Cisco recommends that you grant permission for ICMP unreachable message type (type 3). Denying ICMP unreachable messages disables ICMP Path MTU discovery, which can halt IPSec and PPTP traffic. See RFC 1195 and RFC 1435 for details about Path MTU Discovery."

Since I am using my PIX for IPSEC, I will leave this turned on, but only to my PIX outside Interface IP address, which is my VPN tunnel interface.

access-list 101 deny ip any any

Even though Cisco ACL's have an implicit deny, I'm going to add this at the end just to watch the counter climb. I could add "log" at the end of this rule to log everything that is denied.

access-list 102 permit ip XXX.240.1.0 0.0.0.31 any
access-list 102 permit icmp XXX.240.1.0 0.0.0.31 any packet-too-big
access-list 102 permit icmp host XXX.240.1.29 any echo
access-list 102 permit icmp host XXX.240.1.29 any host-unreachable log
access-list 102 deny icmp any any
access-list 102 deny ip any any log

This is my Egress filter which is applied to Interface ethernet 0/2 to outbound traffic. I am blocking outbound ip traffic that is not internal to my border router, which will only be the X.240.1.0/27 subnet that my ISP assigned me. Since my Cisco PIX will be NAT'ing all addresses, there is no reason for any other addresses to show up on this router as the originating IP address going outbound through this interface. And if by chance they would, the "deny ip any any" statement at the very end would block it. I'm blocking all ICMP outbound packets except for packet-too big, host-unreachables and echo requests. In the case of host-unreachable and echo requests, I am only permitting this from My Cisco PIX outside Interface for reasons that I stated earlier for access list 101.

access-list 103 permit udp host XXX.240.1.17 any eq domain
access-list 103 permit tcp host XXX.240.1.17 eq telnet host XXX.240.1.18
access-list 103 permit udp host 1 XXX 0.240.1.17 eq domain host XXX.240.1.18
access-list 103 deny ip host XXX.240.1.17 any log

I explained this access list earlier when I was discussing it as it was applied to interface ethernet 0/1. As a quick refresher, this list allows my Internet Visible DNS to initiate DNS queries to the Internet. It allows me to telnet to the DNS from my Border Router. And it allows me to make DNS queries to the DNS from the Border Router. In case you were wondering, since my Internal Network already knows that X.240.1.17 is on the Internal network, I can only telnet to my Internet Visible DNS by telnetting to my Border Router, authenticating, then telnetting to X.240.1.17 from there. The best security practice would not give me the ability to remotely logon to my Border Router, but I'll have to state a business case on this one. I'm not always within walking distance of the Internet Visible DNS.

access-list 109 permit ip XXX.240.1.0 0.0.0.7 any (Service Net 1 Subnet)
access-list 109 permit ip XXX.240.1.28 0.0.0.3 any (PIX Outside Interface Subnet which is our VPN tunnel IP Address)
access-list 110 permit ip XXX.240.1.20 0.0.0.7 any (PIX NAT and PAT Subnet)

These access lists are used for traffic shaping and were discussed earlier under interface ethernet 0/2.

banner motd _
****

GIAC Enterprises Gateway
WARNING: This system accesses proprietary information.
Access is restricted to authorized users only for legitimate Corporate
Business purposes. Unauthorized access may be a violation of state

and federal, civil and criminal laws and is subject to monitoring.
****

The banner motd is displayed to all connected terminals and should serve as a warning that intruders are not welcome. I found an FBI site at http://www.fbi.gov/programs/ipcis/ipcis.htm which states:
"consider placing a warning banner on your system to notify unauthorized users they may be subject to monitoring and data residing on the system is subject to review"
Syntax: **banner motd** *d message d,* where the *d* is the delimiting character.
You cannot use the delimiting character in the banner message. I used an underscore "_".

line con 0
line aux 0

No special configuration for the console or auxiliary ports.

line vty 0 4
access-class 1 in
password xxxx (x'ed out)
login

If I absolutely didn't want anyone to be able to remotely login to this router, I would put in the "no login" command. But since I do want to be able to login remotely, I added a command that only allows permitted IP addresses access. In this case, access-class 1 in refers to access list 1, which permits the NAT IP address of my syslog server. This allows anyone that has access to the syslog server to telnet to the Border Router. The password is displayed in encrypted form due to the previously set "service password-encryption" command. If this were not set, the password would be displayed in clear text.

ntp clock-period 17179610

Caution Do not enter this command; it is documented for informational purposes only. The system automatically generates this command as Network Time Protocol (NTP) determines the clock error and compensates.

ntp server XXX.240.1.22

This is the NAT IP address of my Internal NTP Server.
Syntax: ntp server *ip-address* [version *number*] [key *keyid*] [source *interface*] [prefer]
The version is defaulted to 3. I do not use a key for authentication, although that would help ensure that I'm actually getting my time from the real NTP server and not one that is spoofing the IP address. I haven't specified a source interface since I know that it will always use the interface that the traffic is exiting the router on going to the NTP server. I also haven't set "prefer", which is a way to set a preference if I had multiple NTP servers configured.

PIX Version 5.3(1)
nameif ethernet0 outside security0
nameif ethernet1 inside security100
nameif ethernet2 servicenet1 security50
nameif ethernet3 servicenet2 security75

The nameif command lets you assign a name to an physical interface. The security levels assigned to each interface are used to decide whether traffic is initially permitted or initially denied. Traffic flowing from a network on an interface with a higher security level to a lower one is initially permitted, unless specifically blocked. Traffic flowing from a network on an interface with a lower security level to a higher one is initially denied unless otherwise permitted by a conduit or access list. It looks like Cisco wants everyone to start using access lists instead of conduits, so that it what we'll do.
Syntax: nameif *hardware_id if_name security_level*

enable password xxxxxxxx encrypted (x'ed out)

This is the password that is used to enter the privileged mode. It can be up to 16 alphanumeric characters. Encrypted just means that you are viewing the password in encrypted form.
Syntax: enable password*password* [encrypted]
You would only use the encrypted command if you were entering the password in it's encrypted form. You might do this if you were recreating your configuration from a backup file.

passwd xxxxxxx encrypted (x'ed out)

This command is used to set the telnet password for access to the console. It is also displayed in it's encrypted form.
Syntax: passwd *password* [encrypted]
You would only use the encrypted command if you were entering the password in it's encrypted form.

hostname giacpix

Sets the host name for command prompts.
Syntax: hostname *newname*
Cisco warns that if you are using Digital Certificates, that changing the hostname changes the fully qualified domain name, and that you should delete your RSA key pairs and Digital Certificate.

domain-name 4chewns.com

This command is used when generating RSA key pairs. We will not be using Digital Certificates, but there is no harm setting it.

fixup protocol ftp 21
fixup protocol http 80
fixup protocol h323 1720
fixup protocol rsh 514
fixup protocol smtp 25
fixup protocol sqlnet 1521
fixup protocol sip 5060

The fixup protocol commands are part of the PIX Firewall application protocol feature. By turning this on for each application, it performs the Adaptive Security Algorithm based on different port numbers other than the defaults. This command is global, and changes things for both inbound and outbound connections, and cannot be restricted to any static command statements.
For http, this enables the ability to filter URL's.
In other words, it knows something about each application. An example of this may be that you initiate a request on port 514, but the remote server may want to initiate a session back to you on several TCP or UDP ports. The fixup command enables the other ports based on what it knows about the behavior of the application that normally uses that port and makes allowances for it. For other protocols, it may simply watch the packets, and only allow well known, or allowed commands. For me to explain each one would simply be a lot of cutting and pasting. You should read them for yourself at http://www.cisco.com/univercd/cc/td/doc/product/iaabu/pix/pix_v53/config/commands.htm#xtocid223322.

names

This command allows you the ability to use the "name" command, which lets you associate an IP Address with a hostname. It would be useful for naming your tftp server, or a device that you normally ping for troubleshooting.

access-list vpnclientbypassnat permit ip YYY.1.1.0 255.255.255.0 YYY.1.2.0 255.255.255.128
access-list vpnclientbypassnat permit ip XXX.240.1.0 255.255.255.224 YYY.1.2.0 255.255.255.128

When applied by a VPN rule this access list will allow all of the IP addresses that are assigned to VPN Clients from the pool "vpnclients" to access all of our Internal IP addresses, as well as those on the different interfaces of our firewall and our Internal DNS. When applied by a nat 0 rule, it allows any traffic that matches to bypass the NAT or PAT rules that are normally configured for the Firewall.

access-list partner1 permit tcp host XXX.240.1.24 eq telnet PPP.114.170.8 255.255.255.0

When applied to a VPN rule, this access list will allow the entire subnet at our partner1 site to telnet to our development server's NAT address.

access-list supplier1 permit ip host XXX.240.1.8 host SSS.1.2.200

When applied by a VPN rule, this access list will allow the host listed at one of our suppliers to have unlimited access to the database server in servicenet2.

access-list acl_outside permit tcp any host XXX.240.1.1 eq 443

Permits SSL access to my web server from the Internet.

access-list acl_outside permit tcp any host XXX.240.1.2 eq smtp

Permits smtp mail to be received from the Internet.

access-list acl_outside permit udp host XXX.240.1.30 host XXX.240.1.21 eq syslog

Permits syslogs from the Border Router to the Internal syslog server.

access-list acl_outside permit udp host XXX.240.1.30 host XXX.240.1.22 eq ntp

Permits the Border Router (via the ip of the ethernet 0/0 interface) to make Network Time requests.

access-list acl_outside permit udp host XXX.240.1.30 host XXX.240.1.23 eq tftp

Permits configuration backup via TFTP for the Border Router.

access-list acl_svcnet1 permit ip host XXX.240.1.1 host XXX.240.1.9 eq 11111

Permits the web server to make a query to the database server on servicenet2 on tcp port 11111.

access-list acl_svcnet1 permit XXX.240.1.0 255.255.255.248 host YYY.1.1.4 eq smtp

Permits email that is accepted at the mailrelay server from the Internet and email that may be generated by any other server in ServiceNet1 to be sent to the internal mail server.

access-list acl_svcnet1 permit udp host XXX.240.1.1 host YYY.1.1.4 eq domain
access-list acl_svcnet1 permit udp host XXX.240.1.2 host YYY.1.1.4 eq domain

Permits the mailrelay server and the WWW server to use the Internal DNS. I have it using the Internal DNS for two reasons. #1, I want to be able to resolve internal addresses properly. #2, I don't want to use a DNS that could possibly have it's cache poisoned from the Internet, which could potentially point email destined for my internal mail server out to an external server.

access-list acl_svcnet1 permit tcp host XXX.240.1.4 any eq smtp

I allow the mailrelay server to send email anywhere on the Internet. The server itself is configured to only relay mail to or from our domain.

access-list acl_svcnet1 permit udp host XXX.240.1.3 any eq domain

I allow the DNS server to make DNS request to anywhere on the Internet.

access-list acl_svcnet2 permit udp host XXX.240.1.9 host XXX.240.1.17 eq domain

Allow the db server to do DNS queries against the Internal DNS.

pager lines 24

The pager lines command lets you specify the number of lines in a page before the More prompt appears.

logging on

Enables the sending of syslog messages to all output locations.

no logging timestamp

If turned on, each message would be sent to the syslog server with a timestamp.

no logging standby

If you are configured with another PIX in the Failover configuration, it will make the standby unit send syslog messages too. If used, this causes both PIX units to send the same syslogs at the same time while both are working. If one fails, you would of course only get logging from one. This is supposed to prevent you from losing any syslog data in the event of a failover.

no logging console

Cisco suggests that you leave this turned off due to performance issues It also makes it tough to login from the console and work if you have tons of syslog messages screaming by.

no logging monitor

If you wanted to view syslog messages via a telnet connection, you would enter the logging monitor command. This is useful when running a debug. Don't forget to turn it off when your done.

logging buffered notifications

Sends syslogs to the buffer, which can be viewed by the "show log" command. In this instance, I am allowing notification messages and lower to be written to the buffer. Here's the levels that you can view:
0—emergencies—System unusable messages
1—alerts—Take immediate action
2—critical—Critical condition
3—errors—Error message
4—warnings—Warning message
5—notifications—Normal but significant condition
6—informational—Information message
7—debugging—Debug messages and log FTP commands and WWW URL's

logging trap debugging

This is the level of the messages that I send to my internal syslog server

no logging history

This is for sending SNMP traps, which I do not have turned on.

logging facility 23

The logging facility that is in use by the syslog server.

logging queue 1000

Specifies the amount of messages held in the queue before being sent to the syslog server.

logging host inside YYY.1.1.1

The IP address of my syslog server. You could list multiple servers if you want to.

interface ethernet0 10baset
interface ethernet1 10baset
interface ethernet2 10baset
interface ethernet3 10baset

I set the interface speed to 10baset. The "auto" keyword can only be used with the Intel 10/100 automatic speed sensing network interface card. Cisco recommends that you do not use the "auto" option to maintain compatibility with switches and other devices in your network.

mtu outside 1500
mtu inside 1500
mtu servicenet1 1500
mtu servicenet2 1500

The ethernet MTU is 1500, so that is what we set this for. I've seen some mention in news groups that setting this to 1400 will help fix some VPN problems. I haven't had to try that.

ip address outside XXX.240.1.29 255.255.255.252
ip address inside YYY.1.1.9 255.255.255.248
ip address servicenet1 XXX.240.1.6 255.255.255.248
ip address servicenet2 XXX.240.1.14 255.255.255.248

This is where we assign the ip address to the different interfaces.

ip audit info action alarm

ip audit attack action alarm

"Cisco Secure Intrusion Detection System (Cisco Secure IDS) is an IP-only feature that provides some level of flexibility for the user to customize the amount of traffic that needs to be audited and logged"

These commands are fairly new to this version. I have not had a chance to play with any of them yet, but it looks like it is an attempt to weed out what is just normal denied traffic and what may be an actual attack. This is done by watching for known signatures within the packets. This also gives you the ability to take certain action based on these attacks.

ip local pool vpnclients YYY.1.2.1-YYY.1.2.126

This is a pool of IP addresses that are dynamically assigned to VPN Clients. InternalRouter1 must have a static route built to the inside interface of the PIX for this network.

no failover
failover timeout 0:00:00
failover poll 15
failover ip address outside 0.0.0.0
failover ip address inside 0.0.0.0
failover ip address servicenet1 0.0.0.0
failover ip address servicenet2 0.0.0.0

This is where you would set the failover parameters for a failover PIX unit. The interfaces must be connected and addressed to the same subnets as the primary unit. I am not defining a failover solution in this practical.

arp timeout 14400

This is the arp cache timeout. Hardcoded entries can be made. But if you have your subnets configured correctly and are not overlapping, the PIX handles answering ARP requests OK.

global (outside) 1 XXX.240.1.25 netmask 255.255.255.248

This is where I specify the PAT address that is to be used. I could specify a range of addresses to use for dynamic NAT, but we're short on the addresses available. Outside is the interface. The number 1 specifies the number that is shared between the global and NAT commands. You will see in a few moments that any traffic that is matched with the number 1 NAT command and is seen on the outside interface is given the PAT address as assigned by this global statement.

global (servicenet1) 1 XXX.240.1.5 netmask 255.255.255.248

This statement is similar to the one above, but is the PAT address that would be used for the same traffic that matches the number 1 NAT command and is seen on the servicenet1 interface.

nat (inside) 0 access-list vpnclientbypassnat

This statement is added to allow any traffic that matches entries in the access list vpnclientbypassnat, to bypass any global and nat statements that are defined. This is how you get your vpnclient address pool to work for you so that your VPN clients are actually connecting to your real internal IP addresses.

nat (inside) 1 0.0.0.0 0.0.0.0 0 0

This statement says that traffic from any IP address that hits the inside interface is to be applied to the matching global rule 1 for the interface that it would be exiting. I'm not particularly worried about allowing spoofed addresses, since this rule will force it to use the PAT address when going out to the Internet. So it's not like I'm being a bad net citizen.

nat (servicenet1) 1 XXX.240.1.0 255.255.255.248 100 50

This statement says that any traffic that originates in servicenet1 should be applied to the same global rule 1.

static (inside,servicenet1) YYY.1.1.0 YYY.1.1.0 netmask 255.255.255.0 100 50

This command nullifies the earlier command that would have made all my internal traffic get a PAT address when connecting to anything within servicenet1. Now all of my internal IP addresses are visible on the servicenet1 network. This actually makes all of the real IP addresses shared between both of these interfaces visible.

static (inside,servicenet1) XXX.240.1.16 XXX.240.1.16 netmask 255.255.255.252 100 50

This command is used so that the real IP address of the Internal DNS can be seen in the Service Net 1 network.

static (servicenet1,outside) XXX.240.1.0 XXX.240.1.0 netmask 255.255.255.248 100 50

This command nullifies the earlier command that would have made all my servicenet1 traffic use the PAT address when leaving the outside interface. Now, all of the servicenet1 IP addresses are visible to the Internet.

static (inside,outside) XXX.240.1.21 YYY.1.1.1 netmask 255.255.255.255 100 50

The static NAT command for my internal syslog server. Records syslogs from the Border Router.

static (inside,outside) XXX.240.1.22 YYY.1.1.2 netmask 255.255.255.255 100 50
The static NAT command for my internal NTP server. Provides Network Time to the Border Router.

static (inside,outside) XXX.240.1.23 YYY.1.1.3 netmask 255.255.255.255 100 50
The static NAT command for my internal TFTP server. Provides configuration backup via TFTP for the Border Router.

static (inside,outside) XXX.240.1.24 YYY.1.1.5 netmask 255.255.255.255 100 50
The static NAT command for my internal development server.

access-group acl_outside in interface outside
This is where you apply "access-list acl_outside" to the outside interface.

access-group acl_svcnet1 in interface servicenet1
This is where you apply "access-list acl_sevicenet1" to the servicenet1 interface.

access-group acl_svcnet2 in interface servicenet2
This is where you apply "access-list acl_servicenet2" to the servicenet2 interface.

route outside 0.0.0.0 0.0.0.0 XXX.240.1.30 1
This is the default route statement. It defines that traffic destined for IP addresses not otherwise specified should go to the Border Router.

route inside YYY.1.1.0 255.255.255.0 YYY.1.1.14 1
This static route statement tells traffic destined for the YYY.1.1.0/24 network to go to Internalrouter1.

route inside XXX.240.1.16 255.255.255.252 XXX.1.1.14 1
This statement allows traffic to route to and from the Internal DNS. The Internal DNS is configured with forwarders statements to query the DNS on servicenet1 for all domains other than 4chewns.com. If it would happen to directly connect to the Internet, it would pick up the PAT address of XXX.240.1.25 because it matched the earlier defined nat and global rules as applied to the Internal and outside interfaces.

timeout xlate 3:00:00
This is the maximum idle time that a NAT or PAT connection can have until the resource is returned back to the pool. The default is 3 hours. It can be set for as little as one minute.

timeout conn 1:00:00 half-closed 0:10:00 udp 0:02:00 rpc 0:10:00 h323 0:05:00 sip 0:30:00 sip_media 0:02:00
"timeout connection" is the maximum idle time of a particular connection until the slot is freed. Must be at least 5 minutes. The default is one hour.
"half-closed" is the maximum time that a half open connection is maintained. The default is 10 minutes. The minimum is 5 minutes.
"udp" is the maximum idle time that a a udp connection is held open. Minimum is one minute. The default is 2 minutes.
"rpc" is the maximum idle time until an rpc connection is freed. Minimum is 1 minute. The default is 10 minutes.
"h323" is the maximum idle time until and h323 sessions is freed. Duration must be at least 5 minutes.
"sip" is the maximum idle time for the sip ports. This is defaulted to 30 minutes.
"sip_media" is the maximum idle time for udp ports used for sip. It is defaulted to 2 minutes.

timeout uauth 0:10:00 absolute uauth 0:05:00 inactivity
"timeout uauth" is the "absolute" amount of time that authentication cache is active. This must be set to a number higher than the inactivity. I set this to 10 minutes to allow the VPN Clients the ability to open multiple tunnels, if required, within 10 minutes of authentication, without authenticating again. After that time, any new tunnel requests would be prompted for their authentication credentials again..
"inactivity" If the session becomes idle after 5 minutes, force authentiation again even though we haven't reached the 10 minute absolute value.

aaa-server authenticate protocol tacacs+
This defines "authenticate" as a group which will be used as a matching rule for authenticartion methods. tacacs+ is the defined authentication server type that we are using.

aaa-server authenticate (inside) host YYY.1.1.6 XXXXXX timeout 5 (TACACS password X'ed out)

This is using the same group "authenticate", and telling it to talk to server YYY.1.1.6 via the inside interface using tacacs password XXXXXX. The timeout for a response is 5 seconds. You may define multiple servers per group. I didn't put this on the diagram just to save some space.

aaa authentication telnet console authenticate

This command tells the firewall to use the group "authenticate" as the method used for the PIX administrator to authenticate to the PIX via telnet. Telnet is not the recommended approach for administering your firewall since all the traffic is in the clear, but we'll show it here. SSH access is the recommended method for administrator access.

no snmp-server location
no snmp-server contact
snmp-server community dfaw;flqwhf
no snmp-server enable traps

I do not use SNMP on the firewall. The no option was not available under community, so I changed it to something other than public or private just to be safe.

tftp-server inside YYY.1.1.3 giacpix

This is the interface, IP address and filename that will be used when running the "write network" command, which saves your configuration file to the tftp server.

floodguard enable

This command is on by default and helps reclaim uauth resources if they are under attack and becoming depleted. If the PIX Firewall uauth subsystem is depleted, TCP user resources in different states are reclaimed depending on urgency in the following order:

1. Timewait
2. FinWait
3. Embryonic
4. Idle

sysopt connection permit-ipsec

This command permits IPSEC traffic to traverse the firewall without being checked by access lists or conduits. This command is added so that I do not have to create access lists for every internal host that my VPN clients may want to connect to.

no sysopt route dnat

"Specify that when an incoming packet does a route lookup, the incoming interface is used to determine which interface the packet should go to, and which is the next hop."
This sounds straight forward, but since it was not on by default, I'm going to leave it off.

auth-prompt prompt WARNING: GIAC Enterprises Gateway: Authorized Users Only!

This is the prompt provide to all users that may authenticate. This is not displayed to adminstrators when they log in via telnet. This is what users would see if we were authenticating users via telnet, ftp or http when trying to connect to a server through the firewall. I am not doing this in this pratical.

crypto ipsec transform-set vpntrans esp-3des esp-md5-hmac

This defines the IPSEC transform set and it's parameters. We will be applying this transform set to our VPN Clients. esp means we are using Encapsulated Payload, which means we are going to encrypt the entire data portion of the packet.
3des means we are using triple DES encryption, 168 bit.
esp-md5-hmac means that the md5 hash is used to authenticate the esp. HMAC is a variant which I cannot find a good explanation of.

crypto ipsec transform-set suppliertrans esp-3des esp-md5-hmac

Our supplier is in the US, so they are capable of 3DES also.

crypto ipsec transform-set partnertrans esp-des esp-md5-hmac

Since we are working with an international partner we've decided to stick with single DES encryption for this connection.

crypto dynamic-map vpnclientmap 10 set transform-set vpntrans

This map alows us to accept requests for new security associations from previously unknown peers. This is configured since we may not know what IP address most of our VPN Clients will be coming from.

crypto map allmap 25 ipsec-isakmp
crypto map allmap 25 match address supplier1
crypto map allmap 25 set peer SSS.107.146.5
crypto map allmap 25 set transform-set suppliertrans

This is the rule that is used to decide how to handle the VPN connection for our supplier.
ipsec-isakmp is used to state that we are using isakmp negotiation
The match address command tells it what access list to look at for valid traffic that is allowed to build the IPSEC tunnel.
The peer address is the IP address of the partners VPN gateway.
The transform set tells it what transforms to use for negotiating the tunnel.

crypto map allmap 26 ipsec-isakmp
crypto map allmap 26 match address partner1
crypto map allmap 26 set peer PPP.114.66.5
crypto map allmap 26 set transform-set partner1trans

This is basically the same information, except for our partner.

crypto map allmap 30 ipsec-isakmp dynamic vpnclientmap

This is the crypto map statement that is used for the VPN Clients. It knows to use this by the previously assigned dynamic-map, vpnclientmap. Notice how this number is higher than the previously listed crypto map statements. They are checked in order and this is the last one to be checked.

crypto map allmap client configuration address initiate

The PIX firewall will attempt to initiate the IP address for the peer.

crypto map allmap client authentication authenticate

Tells the client that the authentication method is "authenticate", which is the "aaa-server authenticate protocol tacacs+" command that was entered earlier.

crypto map allmap interface outside

Which interface the PIX uses to identify itself to other IPSEC peers (gateways or clients).

isakmp enable outside

Enables isakmp negotiation on this interface.

isakmp key XXXXXXXXX address PPP.114.66.5 netmask 255.255.255.255 no-xauth no-config-mode

Specifies our partners IPSEC gateway IP address and the preshared key to be used. Disables xauth authentication (Like the VPN Clients use) and disables the config mode, which would allow the peers to be assigned or assign an IP address for the connection.

isakmp key XXXXXXXXX address SSS.107.146.5 netmask 255.255.255.255 no-xauth no-config-mode

Basically the same info as above except this is for our supplier.

isakmp identity address

Specifies that we are authenticating users by IP Address. Like we have defined in the isakmp key rules above.

isakmp client configuration address-pool local vpnclients outside

VPN Clients will use the "ip local pool vpnclients YYY.1.2.1-YYY.1.2.126" to get a virtual IP address assigned.

isakmp policy 10 authentication pre-share

State that we are using preshared keys for authentication

isakmp policy 10 encryption des

We are only using DES encryption during the isakmp negotiation.

isakmp policy 10 hash md5

Using MD5 as the one way hash

isakmp policy 10 group 1

Using Diffie-Hillman group 1 (768 bit)

isakmp policy 10 lifetime 86400

isakmp lifetime is good for 24 hours. After this time a new isakmp negotiation is required to continue. This will cause a VPN Client to reauthenticate seemingly in the middle of working on something.

vpngroup giac_employee address-pool vpnclients

This is part of the new VPN3000 Client config. This command and those that follow allow me to define different parameters for different groups. I really like this because I can hand out different shared secret keys to different groups of people. There is an ipsecdlr.ini file that you create that the end user imports into the VPN3000 Client. It specifies the groupname, the shared secret and the IP address of the outside interface of the PIX. The rest of the information is sent down to the client during the isakmp negotiation. The groupname applies an access list (vpnclientbypassnat) that was created earlier in this configuration which defines the actual traffic that is going to be encrypted in the VPN tunnel.

vpngroup giac _employee dns-server XXX.240.1.17

Downloads to the VPN Client the domain name servers ip address. On Win95 PC's this will actually show up in your winipcfg.

vpngroup giac _employee default-domain 4chewns.com

Downloads to the VPN Client the domain name. On Win95 PC's this will actually show up in your winipcfg.

vpngroup giac _employee split-tunnel vpnclientbypassnat

Permits the VPN Client user to access the Internet directly without connecting to us via the VPN and then getting sent back out again.

vpngroup giac _employee idle-time 1800

vpngroup giac _employee password coqwchvo5voerhg453werhvvrr

The shared secret for this groupname.

telnet YYY.240.1.1 255.255.255.255 inside

Allows my syslog server to telnet to the PIX for administration. This isn't the best security since all the traffic is in the clear, but at least it's restricted to one IP address.

telnet timeout 15

Idle timeout the administration telnet session

ssh YYY.240.1.1 255.255.255.255 inside
ssh timeout 5

Same commands as telnet above, but with SSH. This is the preferred method for administrator access to the PIX since it is encrypted.

terminal width 80

Terminal display width is default to 80 characters.

Assignment

3

Audit Your Security Architecture

You have been assigned to provide technical support for a comprehensive information systems audit for GIAC Enterprises. You are required to audit the Border Router and Primary Firewall described in Assignments 1 and 2.

Your assignment is to:

1.Plan the assessment. Describe the technical approach you recommend to assess your perimeter. Be certain to include considerations such as what shift or day you would do the assessment. Estimate costs and level of effort. Identify risks and considerations.

2.Implement the assessment. Validate that the Border Router and Primary Firewall are actually implementing the security policy. Be certain to state exactly how you do this, including the tools and commands used. Include screen shots in your report if possible.

3.Conduct a perimeter analysis. Based on your assessment (and referring to data from your assessment), analyze the perimeter defense and make recommendations for improvements or alternate architectures. Diagrams are strongly recommended for this part of the assignment.

Note: DO NOT simply submit the output of nmap or a similar tool here. It is fine to use any assessment tool you choose, but annotate the output.

1. I will initially assess my Border Router and Firewall configuration as seen from the Internet. There is a spare port (ethernet 0/3) on the Border Router available. I am going to configure it with an IP address and apply the very same Ingress and Egress filters that are applied to ethernet 0/2. This will give me a handy place to connect my laptop and I won't have any ISP's trying to shut me down. It should also speed up my scanning since I'll have a full 10baset ethernet pipe to work with. Since the idea is not for me to be covert about doing this, I have scheduled a network downtime window, lasting approximately 8 hours. Ideally this would be during a weekend or 3rd shift weekday when the Internet connection isn't being used as hard. I really don't plan on taking anything down, but it would be great to let the folks responsible for the servers know. 8 hours should give me plenty of time to do some Nmap scanning of this fairly small subnet from the Internet. Additional time will be required to connect to,and scan from the individual Service netowrks. I'll be spending much more quality time with the sylogs from the Border Router and Firewall later. The syslogs will help me correlate the data that I get back from my scan and where the blocking, or possible breach exists.
2. I'll start by scanning as seen from the Internet and progressively move from one service network to the other and try to breach the perimeter from each of those. For brevity, I will only be showing the screenshots for a couple of my scanning attempts from the Internet perspective and to the specific hosts. In reality I would be scanning all the addresses, even the ones on my router interfaces.
   I picked up a copy of Linux Mandrake for Windows at Staples for $19.95. This allowed me to install a small working version of Linux on my Laptop without unloading Win98 and wiping my hardrive clean. Then I loaded Nmap version 2.5.3 from www.insecure.org/nmap. Now I'm really dangerous.
   I will be sanitizing all the IP addresses used in the screenshots.
   The scans I chose were the TCP SYN Scan and the UDP scan without pinging. I am also dumping the output to a file for future reference. I liked using the command line better than the GUI. I think the GUI was causing a memory leak in my PC.

   I will be showing screen shots from my Nmap output, the Border Router, and Cisco PIX syslogs. The dates and times may not seem to be in any logical order because I had to fit this into my daily work schedule.

Scanning my Internet visible DNS on ServiceNet3.

Nmap

```
# Nmap (V. nmap) scan initiated 2.53 as: nmap -P0 -sU -oN udpscan1 XXX.240.1.17
Interesting ports on guessmy.4chewns.com (XXX.240.1.17):
(The 1447 ports scanned but not shown below are in state: filtered)
Port       State       Service
53/udp     open        domain

# Nmap run completed at Wed Feb 14 13:40:12 2001 -- 1 IP address (1 host up) scanned in 1207 seconds
```

Border Router

Feb 14 13:20:06.038 EST: %SEC-6-IPACCESSLOGP: list 101 denied udp SSS.186.6.2(48706) -> XXX.240.1.17(9), 1 packet

Here's one of many denied packets that are being filtered by my Border Routers Ingress filter. This one happens to be blocking udp port 9.

Feb 14 13:20:11.030 EST: %FW-6-SESS_AUDIT_TRAIL: udp session initiator (SSS.186.6.2:1024) sent 46 bytes -- responder (XXX.240.1.17:53) sent 171 bytes

This is a CBAC doing it's thing for udp port 53, a DNS request.

Nmap

```
# Nmap (V. nmap) scan initiated 2.53 as: nmap -P0 -sS -oN tcpscan1 XXX.240.1.17
All 1523 scanned ports on guessmy.4chewns.com (XXX.240.1.17) are: filtered
# Nmap run completed at Wed Feb 14 13:46:30 2001 -- 1 IP address (1 host up) scanned in 180 seconds
```

Border Router

Feb 14 13:43:45.633 EST: %SEC-6-IPACCESSLOGP: list 101 denied tcp SSS.186.6.2(61808) -> XXX.240.1.17(6145), 1 packet

Here's a denied tcp packet. It's just one of 1523 of these.

Scanning the WWW Server on ServiceNet1

Nmap

```
# Nmap (V. nmap) scan initiated 2.53 as: nmap -P0 -sU -oN udpscan2 XXX.240.1.1
All 1448 scanned ports on www.4chewns.com (XXX.240.1.1) are: filtered
```

```
# Nmap run completed at Thu Feb 15 11:36:42 2001 -- 1 IP address (1 host up) scanned in 1178
seconds
```

The Nmap scan reports that all of the 1448 scanned ports are filtered. This is not only due to the individual udp filter rules in the Border Routers Ingress filter, but the "deny ip any any rule" at the end.

Border Router

Feb 14 13:54:16.599 EST: %SEC-6-IPACCESSLOGP: list 101 denied udp SSS.186.6.2(36454) -> XXX.240.1.1(17), 1 packet

Here's some of what my Border Router blocked due to the Ingress filter. The part of the ingress filter that denied these is this: access-list 111 deny udp any any range 1 20 log, whick blocks all udp ports from 1 through 20. In this case it was udp port 17 that was blocked.

Nmap (V. nmap) scan initiated 2.53 as: nmap -P0 -sU -oN udpscan3 XXX.240.1.1
Interesting ports on (XXX.240.1.1):
(The 28 ports scanned but not shown below are in state: filtered)

Port	State	Service
21/udp	open	ftp
22/udp	open	ssh

and so on, and so on……

Just for kicks I removed the "access-list 101 deny ip any any " rule at the end and scanned it again. This time it came back with all but 28 ports scanned listed as OPEN. The Nmap documentation states that during the UDP scan, if it doesn't get an ICMP port unreachable message, it assumes it's open. I have plenty of PIX syslogs that say these ports weren't really open.

PIX Syslog

Feb 15 14:39:18 GIACPIX.4chewns.com %PIX-4-106019: IP packet from SSS.186.6.2 to XXX.240.9.1, protocol udp received from interface "outside" deny by access-group "acl_outside"

Once I removed the "access-list 101 deny ip any any" rule from the ingress filter, the permitted traffic passed my Border Routers Ingress filter, and my PIX firewall blocked access. Here's just a portion of the PIX syslog for this UDP scan. I think this entry is useless since it doesn't show any port information. Access Lists have replaced Conduit statements in the PIX configurations. The blocked messages created against Conduits gave the port information. It's to bad this isn't the case anymore. Or maybe I'm missing something?

Nmap

Nmap (V. nmap) scan initiated 2.53 as: nmap -P0 -sS -oN tcpscan2 XXX.240.1.1/29
All 1523 scanned ports on (XXX.240.1.1) are: filtered
Interesting ports on www.4chewns.com (XXX.240.1.1):
(The 1522 ports scanned but not shown below are in state: filtered)

Port	State	Service
443/tcp	open	https

Interesting ports on mailrelay.4chewns.com (XXX.240.1.2):
(The 1522 ports scanned but not shown below are in state: filtered)

Port	State	Service
25/tcp	open	smtp

Border Router

Feb 14 15:12:08.225 EST: %SEC-6-IPACCESSLOGP: list 121 denied tcp SSS.186.6.2(58164) -> XXX.240.1.1(6001), 1 packet

This is a denied message from the Border Router that matched the Ingress Filter. Notice the udp port of 6001. This was explicity denied by the rule "access-list 101 deny tcp any any range 6000 6255 log".

PIX Syslog

Feb 14 15:08:13 GIACPIX.4chewns.com %PIX-3-305006: Dst IP is network/broadcast IP, translation creation failed for tcp src outside:205.186.6.2/58164 dst servicenet1: XXX.240.1.0/2241

This is a denied message from the PIX that is caused by a scan of the network address for servicenet1.

PIX Syslog

Feb 14 15:10:12 GIACPIX.4chewns.com %PIX-3-106010: Deny inbound tcp src outside:SSS.186.6.2/45771 dst servicenet1:XXX.240.1.1/1009

This is a denied message from the PIX destined to tcp port 1001 on the WWW Server.

Feb 14 15:17:40 GIACPIX.4chewns.com %PIX-6-302001: Built inbound TCP connection 16041419 for faddr SSS.186.6.2/58161 gaddr XXX.240.1.1/443 laddr XXX.240.1.1/443

Feb 14 15:17:40 GIACPIX.4chewns.com %PIX-6-302002: Teardown TCP connection 16041419 faddr SSS.186.6.2/58161 gaddr XXX.240.1.1/443 laddr XXX.240.1.1/443 duration 0:00:00 bytes 0 (TCP Reset-O)

Feb 14 15:20:11 GIACPIX.4chewns.com %PIX-6-302001: Built inbound TCP connection 16044516 for faddr SSS.186.6.2/58161 gaddr XXX.240.1.2/25 laddr XXX.240.1.2/25

Feb 14 15:20:11 GIACPIX.4chewns.com %PIX-6-302002: Teardown TCP connection 16044516 faddr SSS.186.6.2/58161 gaddr XXX.240.1.2/25 laddr XXX.240.1.2/25 duration 0:00:00 bytes 0 (TCP Reset-O)

These are some permitted connections being allowed though the PIX. Notice the destination ports of 443 and 25 for each of the IP addresses. These are the SSL and smtp connections for the servers in servicenet1.

3. I have to confess, on my first run through I had a rule in my ingress filter that permitted IP packets destined to the servers in servicenet1 that I didn't want the Servers to accept. It looked something like this: access-list 101 permit ip any XXX.240.1.0 0.0.0.7. This was added after all of the SANS Top 10 deny rules, but before the "deny ip any any" rule at the very bottom. Imagine my surprise when Nmap said that I had 28 filtered and the rest were open! After checking the PIX Syslogs and the Nmap documentation for the UDP scan that I was running, I realized that these were false positives. My initial reaction was to just leave it that way since the PIX seemed to be doing the job. I may also have occasion to permit a server in ServiceNet1 to get to the Internet, maybe to get a Digital Certificate assigned. This means that I would have to get into the Border Router and modify that rule, otherwise I would never get a response back to my http request. Then I heard that little voice inside my head saying "defense in depth, defense in depth". So it's just a filter. It is still one more thing that has to be breached before reaching my ServiceNet1 servers. I did compromise on the rules for the smtp server. I created a rule that only allowed replies on ephemeral ports, all those above 1024. **"**access-list 101 permit tcp any eq 25 host XXX.240.1.2 gt 1024 **"**

Another possible recommendation would be to turn CBAC on for the entire Border Router. I'm hesitant to do this on a pipe that could get larger as usage grows. I would have to do some more testing to see if the router could handle all of the extra processing of firewalling everything.

I could also configure tacacs authentication on the Border Router. In that case, even though the administrator could connect from the syslog server's NAT IP address, an individual userid and password (or possibly a token) would be required.

Assignment

4

Design Under Fire

The purpose of this exercise is to help you think about threats to your network and therefore develop a more robust design. Keep in mind that the next certification group will be attacking your architecture!

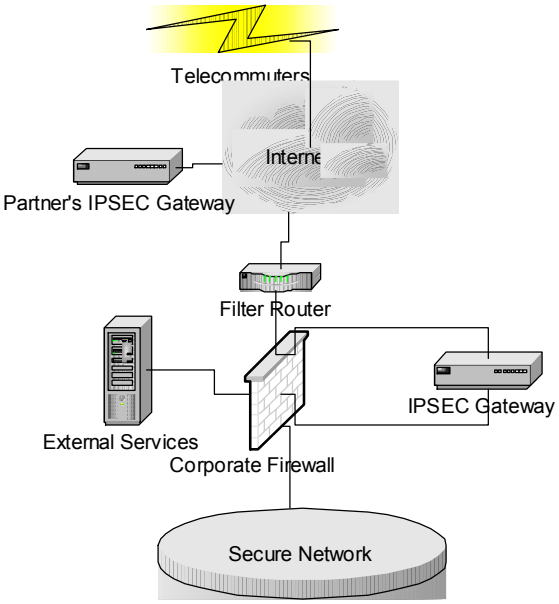
Select a network design from any previously posted GCFW practical (http://www.sans.org/giactc/gcfw.htm) and paste the graphic into your submission. Be certain to list the URL of the practical you are using. Design the following three attacks against the architecture:

1.An attack against the firewall itself. Research vulnerabilities that have been found for the type of firewall chosen for the design. Choose an attack and explain the results of running that attack against the firewall.

2.A denial of service attack. Subject the design to a theoretical attack from 50 compromised cable modem/DSL systems using TCP SYN, UDP, or ICMP floods. Describe the countermeasures that can be put into place to mitigate the attack that you chose.

3.An attack plan to compromise an internal system through the perimeter system. Select a target, explain your reasons for choosing that target, and describe the process to compromise the target.

I have picked Jeremy Browns practical located at http://www.sans.org/y2k/practical/Jeremy_Browns_GCFW.zip. I picked on Jeremy for 2 reasons. 1, he didn't have the highest test scores, and 2, he was the first one listed at the bottom, so it was the first one I viewed. Figure 1 below is a diagram of Jeremy's outermost perimeter protection. Jeremy is using a Cisco PIX firewall. I have checked the Cisco Security Advisory site located at http://www.cisco.com/warp/customer/707/advisory.html and found several vulnerabilities depending on the software version.



1. Depending on the version of the PIX that Jeremy is using there are several attacks that I could try using against the firewall. I do not really see any that would compromise the firewall itself. Most of these either describe a type of denial of service that it is vulnerable to, or an attack against a protected host that it is unable to stop.

Cisco PIX and CBAC Fragmentation Attack

All users of Cisco PIX Firewalls with software versions up to and including 4.2(1) are affected. This has to do with the inability to protect hosts against certain denial of service attacks involving fragmented IP packets. This vulnerability does not permit network "breakins".

Because the firewall drops only the initial fragments of blocked datagrams, attackers can exploit this vulnerability by sending streams of complete fragmented packets. The attacker in this case deliberately intends the initial fragments to be blocked by the firewall. Since only the non-initial fragments will be forwarded, the effect on the target host will be similar to the effect of sending only the non-initial fragments to begin with.

Cisco Secure PIX Firewall TCP Reset Vulnerability

This vulnerability exists in all Cisco Secure PIX Firewall software releases up to and including 4.2(5), 4.4(4), 5.0(3) and 5.1(1). When the Cisco Secure PIX Firewall receives a TCP Reset (RST) packet, it evaluates that packet based on data contained in the TCP packet header: source IP address, source port, destination IP address, and destination port. If these four values match an entry in the stateful inspection table, the associated connection will be reset. This affects only TCP sessions.

This would be a tough vulnerability to exploit because you would need:

Detailed knowledge of the connection table in the Cisco Secure PIX Firewall prior to launching the attack or detailed knowledge of the source and destination IP Address and ports associated with a particular connection to be attacked.

2. My Denial of Service choice is a TCP SYN Flood. Cisco describes the TCP SYN Flood and some defense against it at http://www.cisco.com/warp/customer/707/4.html .

TCP SYN attack: A sender transmits a volume of connections that cannot be completed. This causes the connection queues to fill up, thereby denying service to legitimate TCP users. There is no easy way to trace the originator of the attack because the IP address of the source is forged.

Jeremy's Ingress filter on his border router does not deny any IP addresses that may look like they are from Private addresses, or even his own addresses. I would target his server with packets from all of the compromised cable modem and DSL user that appeared to come from everywhere; Private Addresses, and maybe a few of his own. If it were possible for me to use his server to attack other devices, I would be able to spoof any originating address in the packets since there is no Egress filter to stop me.

There is no 100% method to protect yourself against a TCP SYN Flood attack. By proper application of an Ingress filter on your border router you can prevent TCP SYN Flood packets that may be spoofing originating addresses that you are using or Private addresses. By applying a properly constructed Egress filter you may be preventing a host at your site from becoming a party to a TCP SYN Flood attack by denying any packets that aren't addressed as coming from your own network.

3. I have a few vulnerabilities listed below for various versions of PIX software, but let's assume for this section that the PIX is not running a version affected by this. If I read Jeremy's practical correctly, he made no mention of the need for a firewall or firewall software on the PC's for the telecommuters. IPSEC will hide your traffic, but you can see the gateway and client IP addresses. IPSEC traffic can be recognized by looking for IP Protocol 50 (for ESP) and udp port 500 (for the isakmp negotiation) packets. You may not be able to see the data in the packet, but now you know where the end points are. . I believe my first attempt may be to run Nmap against static IP addresses or maybe a subnet of IP addresses that are known to be VPN telecommuters. Once I have discovered the client vulnerabilities, it may be possible to crack some passwords with l0phtcrack (http://www.securitysoftwaretech.com/l0phtcrack/). Who knows, they may even have PCAnywhere running with no passwords. Once a VPN'ed PC is compromised, I don't have to worry about firewalls anymore because I'm **V**irtually on the **P**rivate **N**etwork.

Cisco Secure PIX Firewall Mailguard Vulnerability

All users of Cisco Secure PIX Firewalls with software versions up to and including 4.4(6), 5.0(3), 5.1(3) and 5.2(2) that provide access to SMTP Mail services are at risk.
The behavior is a failure of the command **fixup protocol smtp [portnum]**, which is enabled by default on the Cisco Secure PIX Firewall.
To exploit this vulnerability, attackers must be able to make connections to an SMTP mail server protected by the PIX.

Cisco Secure PIX Firewall FTP Vulnerabilities

All users of Cisco Secure PIX Firewalls with software versions up to and including 4.2(5), 4.4(4), and 5.0(3) that provide access to FTP services are at risk from both vulnerabilities.
Cisco Secure PIX Firewall with software version 5.1(1) is affected by the second vulnerability only.
The behavior is due to the command **fixup protocol ftp** [**portnum**], which is enabled by default on the Cisco Secure PIX Firewall.