A Real-World Example of Securing a High-Availability E-Commerce Site

GCFW Practical Assignment Version 1.6
for GIAC Certification in
Firewalls, Perimeter Protection, and VPNs

Chris Russell

November 5, 2001

Table of Contents

1 INTRODUCTION

This paper describes a real-world example of a high-availability network designed for secure e-commerce and illustrates how design compromises sneak in at the last minute when balancing cost and scalability vs. security.

The HA network presented here is a collage of similar networks I've encountered (and even designed and built!) for managed application services and e-commerce sites, as well as high-availability internal network infrastructure.

1.1 DISCLAIMER

At the time of this writing, I did not have access to any Cisco gear for testing. Therefore, while the IOS examples are correct *in spirit*, they may contain minor syntactical error. Therefore, please interpret the network design as a starting point, requiring further testing, tweaks, and validation prior to implementation.

2 SECURITY ARCHITECTURE

2.1 CUSTOMER REQUIREMENTS

The following is a partial list of requirements from the *Vendor* ("GIAC Enterprises") that governs the design and implementation of an e-commerce site for online *Content* ("fortune cookie sayings") sales and distribution.

1. Content shall be purchased online using a credit card or (for authorized customers of the vendor) a purchase order number, and content shall be immediately downloaded to the customer once the purchase transaction is complete.

2. An application server shall be used to:

   - Dynamically generate the web site, such as "personalizing" the web pages for the specific customer logged in based upon their preferences and purchase history.
   - Perform all commerce functions and transactions, such as content browsing and the shopping cart.
   - Perform all financial transactions, such as charging credit cards.

3. An online database shall be used to store:

   - The content being sold.
   - Confidential customer information, such as credit card numbers, purchase history, preferences, and personal data.

Therefore, the database server must be highly secure to prevent hacking and unauthorized access to the data.

4. A content management system shall be used to upload and release new content, including translations in different languages. This system shall be private and only accessible to authorized content suppliers and partners, but *not* to customers or the Internet at large.

5. The site shall support international sales. Therefore, in order support sales during normal working hours *worldwide*, the commerce system must be available 24x7 and should support high availability, e.g., "five-9s availability" or 99.9995% uptime. (It's always morning *somewhere* in the world.)

6. All network activity on the site shall be monitored with a network-based intrusion detection system (NIDS).

7. The vendor is not a "technology" or "engineering" company; development and administration of this site is outside their core business. Therefore, the site shall be built and operated by an application service provider (ASP).

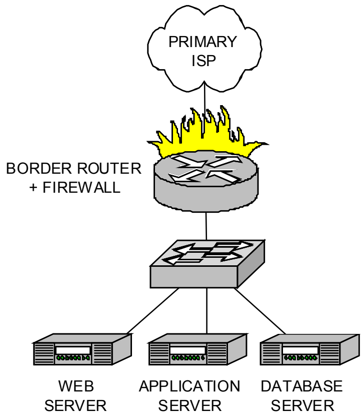
8. Sales are expected to be low for the first few years. Therefore, ASP costs must be competitive and cost effective. As a compromise to lower costs, non-dedicated hardware (shared between multiple vendors that contract the ASP's services) may be used.

2.2 NETWORK DESIGN EVOLUTION

This section describes design process from the ASP's point of view for building a highly secure and available e-commerce system that is also cost-effective and uses shared (non-dedicated) hardware.

2.2.1 Flat Network

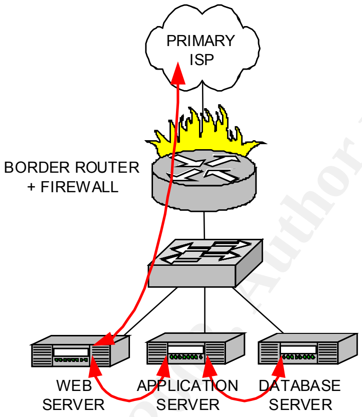
As a starting point, we consider the simplest design first: a single, flat network.



Three servers are required: a web server, application server, and database server. The servers are connected together using a single Ethernet switch to form a flat network, and the network is connected to the Internet through a firewalling border router, as follows:

Only the web server is publicly accessible. The firewall blocks all communication to application server and database server from the Internet.

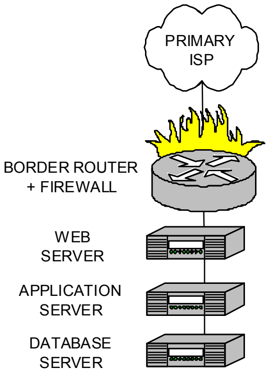
To use the system, a customer connects to the web server, the web server connects to the application server, and the application server connects to the database server.[1] This is the only path of communication between the three servers. In other words, the web server never communicates directly with the database server.



From a performance standpoint, the servers communicate across the switch as if they were directly connected to one another, in a point-to-point topology. However, from a security standpoint, the servers and firewall have full network connectivity with each other, which can be exploited if the firewall or web server were compromised. If link-level connectivity between the servers were limited on a "need to know" basis, then the damage resulting from a security breach could be minimized.

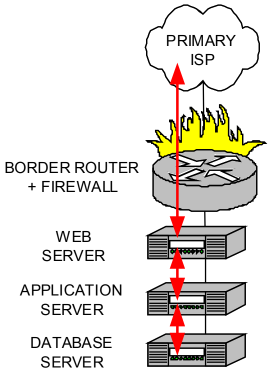
2.2.2 Cascaded Network

In a cascaded network, link connectivity is limited on a "need to know" basis, as follows:



[1] This sentence is sung to same tune as "The ankle bone's connected to the knee bone, …".

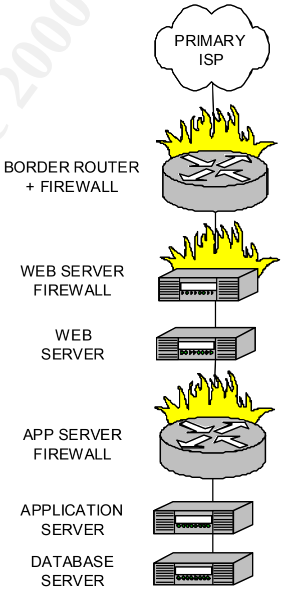
Servers are dual-homed, connected directly to one another, and configured *not* to route packets between their two NICs, thus limiting access and significantly improving security for very little additional cost and configuration. You won't find a better bargain!



For example, if the firewall were compromised, the hacker could only reach the web server. The application and database servers would be completely safe from scanning, detection, network sniffing, and attack. If the web server were compromised, the hacker could then reach the application server but not the database server. Each server provides an additional layer of defense.

2.2.3 Firewalls

The basic e-commerce system is designed using a cascaded network.



The border router is configured to tightly restrict access into the network except to ports 80 (http) and 443 (https) of the web server[2]. However, even with such tight security

[2] The exact security policy for the border router is detailed in §3.1.

settings, firewalling routers (such as the border router) are generally less secure than dedicated firewalls (such as a Cisco PIX or Checkpoint FW-1). For example,

- Most routers do not protect against overlapping packet fragment attacks.
- High-end routers tend to be "feature-rich" and hence more likely to be misconfigured or contain security bugs in their software. Personally, I've battled several router software bugs in my career!

For example, Cisco's documentation for access control lists states: *"Fragmented IP packets, other than the initial fragment, are immediately accepted by any extended IP access list."* [9]

Therefore, a dedicated firewall is installed between the border router and web server. As an added bonus, two layers of firewalls provides additional protection against bugs. If a security bug is discovered for one of the firewalls, hopefully the other firewall does not have the same bug and thus protects the network from being compromised.

The highest security risk for this network is the web server. History is filled with web servers plagued with security bugs, such as buffer overflows, faulty CGI scripts, insufficient validation of user input, poor session security, broken crypto, etc. If the web server is compromised[3], then it may be used to attack the application server and thus compromise the entire system.

Therefore, a simple firewalling router is installed between the web server and application server, thus limiting access from the web server to a bare minimum. Unfortunately, a hacker can still do a lot of damage with this "minimal" access, such as crafting fake remote procedure calls and exploiting vulnerabilities in the backend application software.

If the application server is compromised, then the hacker essentially has complete control over the entire e-commerce system, including the ability to execute custom database operations. Therefore, installing a firewall between the application server and database server would be mostly pointless and thus is avoided.

2.2.4 Active Firewalling

Extra Credit: For added security, it would be ideal for the application firewall to break ***all*** communications between the web server and application server when it detects ***any*** unexpected traffic, such as port scanning, telnet, etc., thus limiting the damage a hacker could inflict and immediately alerting the ASP staff of the intrusion *while* it is in progress.

I do not believe any of the major firewall implementations support this functionality out of the box. **Therefore, it was not incorporated into the initial implemenation of this network.** However, it is possible to construct an active firewall using Linux, ipchains, NIDS software (such as Snort with active response rules), and a couple shell scripts.

[3] Perhaps this should read: "*When* the web server is compromised, …", since it will most assuredly be compromised at some point in the future.

Usually, I (vehemently) advise against configuring firewalls or NIDS systems to actively respond to alerts and shut down connections, since that mechanism may be prone to false alarms or exploited by hackers to trigger a DoS. However, active response inside a cascaded network is the one exception to the rule, for the following reasons:

1. Communications within the cascaded network is well known and tightly controlled. Therefore, it is relatively simple to create rules that will not generate false positives. For example, the following pseudo-code[4] (Cisco style) would shut down a firewall if it encounters any traffic other than web server 10.0.13.11 communicating to app server 10.0.15.11 ports 1000-1010:

   ```
   ip access-list extended filterin
     permit tcp 10.0.13.11 10.0.15.11 range 1000 1010
     deny ip all all trigger interface-down Ethernet 0

   interface ethernet 0
     ip access-group filterin
   ```

   Sniff the link prior to activating the trigger to ensure there are no surprises, such as unanticipated ICMP messages or undocumented ports being used between the two servers. (I've seen it happen!)

2. In order to exploit the trigger to launch a DoS, a hacker first has to gain link connectivity to the firewall or NIDS device. This requires compromising server(s), in this case, the web server. However, once the web server is compromised, the hacker can create a DoS anyway! Plus, he can do lots more damage, such as modifying the web site to intercept customer data (such as logon passwords, new credit card numbers, etc.) or attacking the application server.

2.2.5 Management Server & VPN

Several vendors (including GIAC Enterprises) share the use of these servers. The servers are not dedicated for use by a single vendor. The ASP needs a simple yet secure mechanism for vendors to manage their site, without giving them direct access to the application or database servers. This is accomplished using a management server.

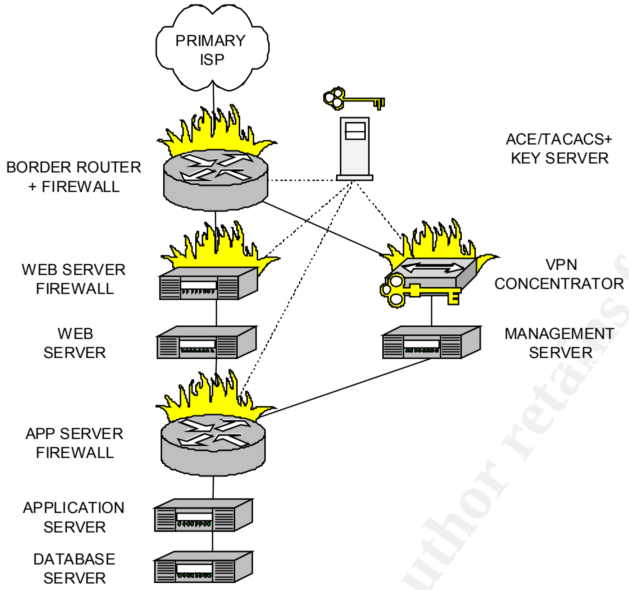
The management server executes account and content management software on the application server, allowing the vendor to access and update their data, user account records (including financial transactions), content, and web site.

For added security, access to the management server is restricted using a secure VPN gateway requiring two-factor authentication. In order to login to the VPN, the customer must use a single-use password generated by entering a private PIN number into a SecurID[5] card. If the wrong PIN is entered multiple times in a row, the card shuts down and must be reset by an administrator. If the single-use password is intercepted by a

[4] Of course, one can come up with an even more powerful and elegant syntax if IOS supports scripting!

[5] Refer to http://www.rsa.com/products/securid for product information.

hacker and reused, then it is automatically rejected. Furthermore, the password is time based, so it is rejected if not used immediately.



The VPN protocol uses IPSEC using ESP encryption authentication. Finally, the management server is connected directly to a separate port on the application firewall (and not connected to the internal web server switch), thus preventing link access from the web servers should they be compromised by a hacker.

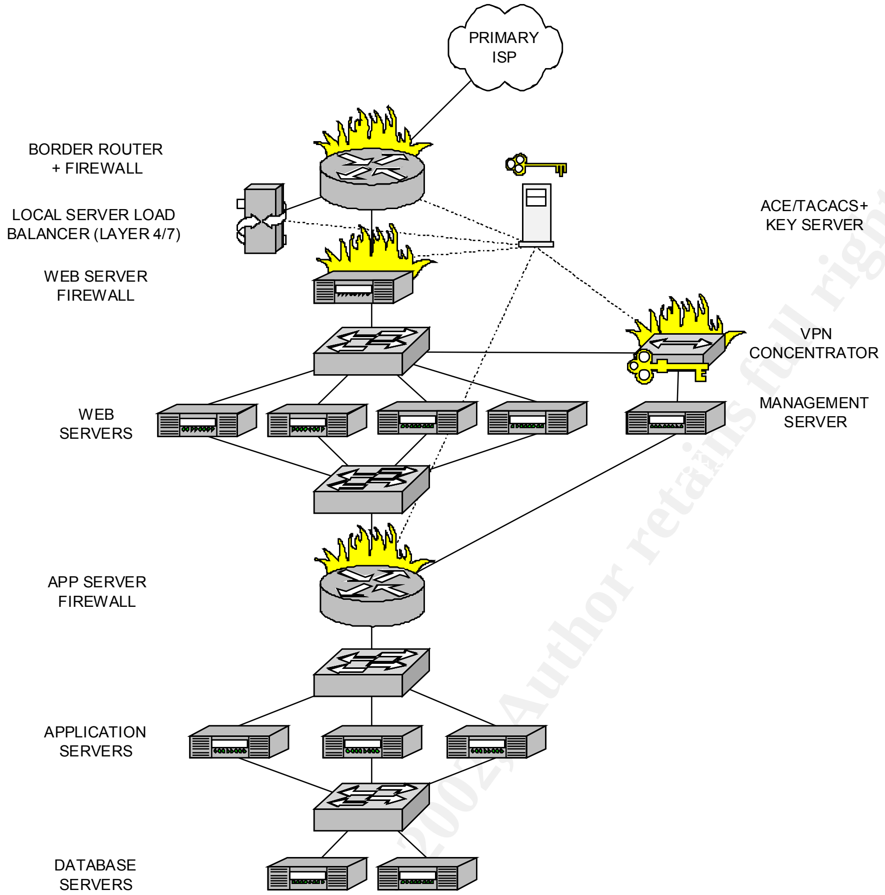
Additional access control may be implemented by the application server, limiting certain vendor accounts to specific functions, such as accounting, account maintenance, upload new content, etc.

2.2.6 Server Redundancy

The customer's requirements specify high availability for the commerce system, which equates to server redundancy.

Incoming traffic is load balanced between multiple web servers using a local server load balancer (LSLB[6]). The LSLB creates a public virtual IP (VIP) address for the set of web servers. When a customer first connects to the VIP using HTTP, the LSLB redirects the connection to one of the web servers, selected either at random, round robin, minimum load, etc. The LSLB can be configured to use cookies (or other mechanisms) to ensure that subsequent HTTP connections from the same user are always redirected to the same web server. The LSLB also maintains a heartbeat with the web servers. If one of the servers goes down, then it is automatically removed from the server pool and will no longer be selected by the LSLB until fixed.

[6] An LSLB is also known as a "layer 4/7 switch" or "content switch". For product examples, refer to http://www.cisco.com/warp/public/44/jump/content_delivery.shtml and http://www.foundrynet.com/products/webswitches/serveriron.



Most application servers[7] use "connection modules" to integrate with the web server. These connection modules typically support automatic load balancing or fail over between multiple application servers. Therefore, no additional hardware is required for fault tolerance of the application servers. That function is already provided in software on the web server.

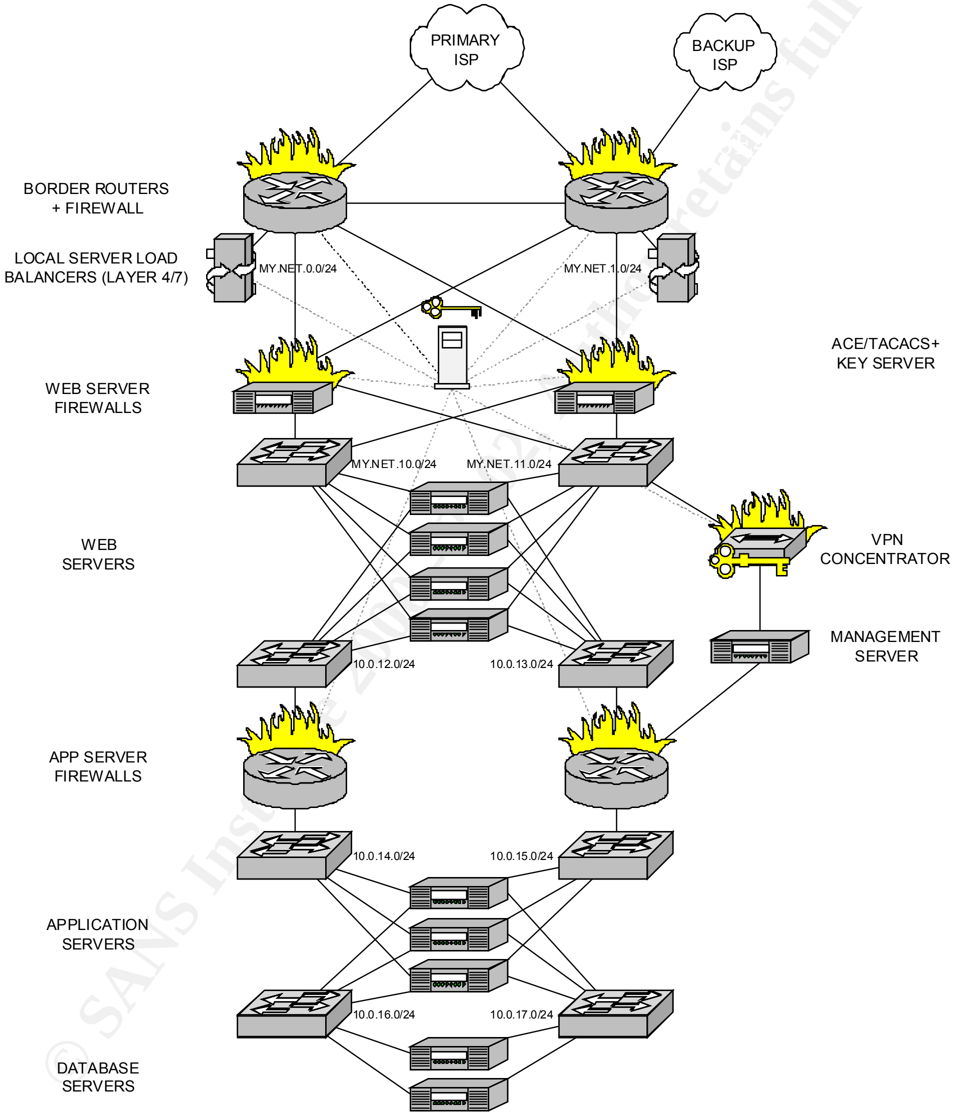
Finally, the database server is made highly available by adding a second, standby server with failover software such as Veritas Cluster Server[8] or Linux-HA[9]. These support (mostly) transparent switchover from the active to the passive server, including MAC and IP address takeover.

Besides the database server, which is active/passive (one device operates while the other is in standby mode), all other servers are operating in fully load-balanced active/active mode, thus making best use of the cost of the systems.

[7] Such as Web Logic, Interstage, and Dynamo, to name a few at random.
[8] Refer to http://www.veritas.com/products/category/ProductDetail.jhtml?productId=clusterserver for product information.
[9] Development on Linux-HA is coordinated at http://www.linux-ha.com. Various Linux releases are now available, including RedHat HA Server (http://www.redhat.com/software/linux/haserver), Mission Critical Linux (http://www.missioncriticallinux.com), Mandrake, etc.

2.2.7 Server & Network Redundancy

Server redundancy protects against the failure of server hardware, such that the system continues to function so long as at least one server of each type is operating: at least one web server, one application server, and one database server. However, if only a single network component fails, such as a router, switch, or firewall, then the entire system goes down in flames. Therefore, for true high availability, the systems requires network redundancy in addition to server redundancy.



Some network technologies were born with redundancy and high availability features since day one, such as FDDI with redundant rings or ATM with redundant links. Unfortunately, Ethernet was not. Therefore, it is fairly complex to build a fault tolerant Ethernet network.

The network presented here uses two parallel networks with systems and key network devices (such as firewalls) redundantly connected to parallel (but different) subnets.

BGP is used on the border router to create redundant, active/active paths (neighbors) to the Internet backbone (primary ISP's autonomous network). Furthermore, a backup ISP is connected to one of the routers, just in case the primary ISP fails altogether.

LSLBs are redundant and typically designed operate active/active with automatic fail over.

Each web server is accessible using two different physical IP addresses, one on each parallel subnet. Each web server appears as two different servers to the LSLB. Thus, since there are 4 physical web server boxes, the LSLB thinks there are really 8 and load balances between them.

> An alternative approach is to set one of the NIC cards in each server to be the primary/active interface and other to be standby/passive. Thus, when the primary interface fails, then the operating system switches over to the standby interface and transfers over the MAC and IP addresses. In theory, this works just as well as previous approach; however, it requires special operating system support, additional configuration, connecting the switches together to create a single subnet, and (in some circumstances) spanning trees, which don't always work as advertised! Therefore, the simpler (a.k.a., more reliable) approach was chosen.

Active/active routing across the web server firewalls is controlled using OSPF (with MD5 authentication, of course!), and link weights are used to sculpt traffic, consistently routing traffic on the left side through the left firewall, and on the right side through the right firewall. If one firewall fails, then it gets all the traffic.

> An alternative approach is to link each pair of switches together (left side + right side pair) to form a single subnet, rather than two different subnets, thus creating redundant network paths within the subnet. Spanning trees disable the redundant links (which would otherwise form loops) and re-enable them as needed to heal from link or network device failures. At least that's in theory. In the past, the implementation of spanning trees has been quirky and therefore was avoided in this network design.

Using redundant network gear, the system should remain operational so long as there is sufficient network connectivity to reach all the networked devices, even if the network path crosses between the left and right subnets.

2.2.8 Finishing Touches

Finally, the following additional security features are added to the network described in §2.2.7 above to complete the ASP's network.

- Advanced QoS is enabled on the border router to limit the flow of certain types of network traffic (such as SYN packets) to make the system resistant to certain DDoS attacks. During an actual attack, the packet filter rules and QoS configurations can be manually tuned on the fly to fight against the specific form of DDoS.

- NIDS probes are installed to monitor network traffic across each of the subnets. (There are a total of 8 subnets.)

- All systems shall execute virus detection software, if available for the given platform. Virus detection strings shall be automatically updated daily (minimally).

- All software changes and patches shall require testing and approval by Change Management prior to being released onto production systems. However, if the software updated deemed to be a critical security patch, then approval will be fast tracked for quick installation, on an as-needed basis.

The resulting network (see diagram in §2.2.7 above) incorporates a number of high-end scalability, availability, and security features, making the network and security engineers very happy. Unfortunately, it is deemed to be too expensive. To be cost effective, the ASP needs to scale it back, somehow.

2.2.9 VLANs & Hardware Reuse

To reduce costs, the ASP implements the network using VLANs.

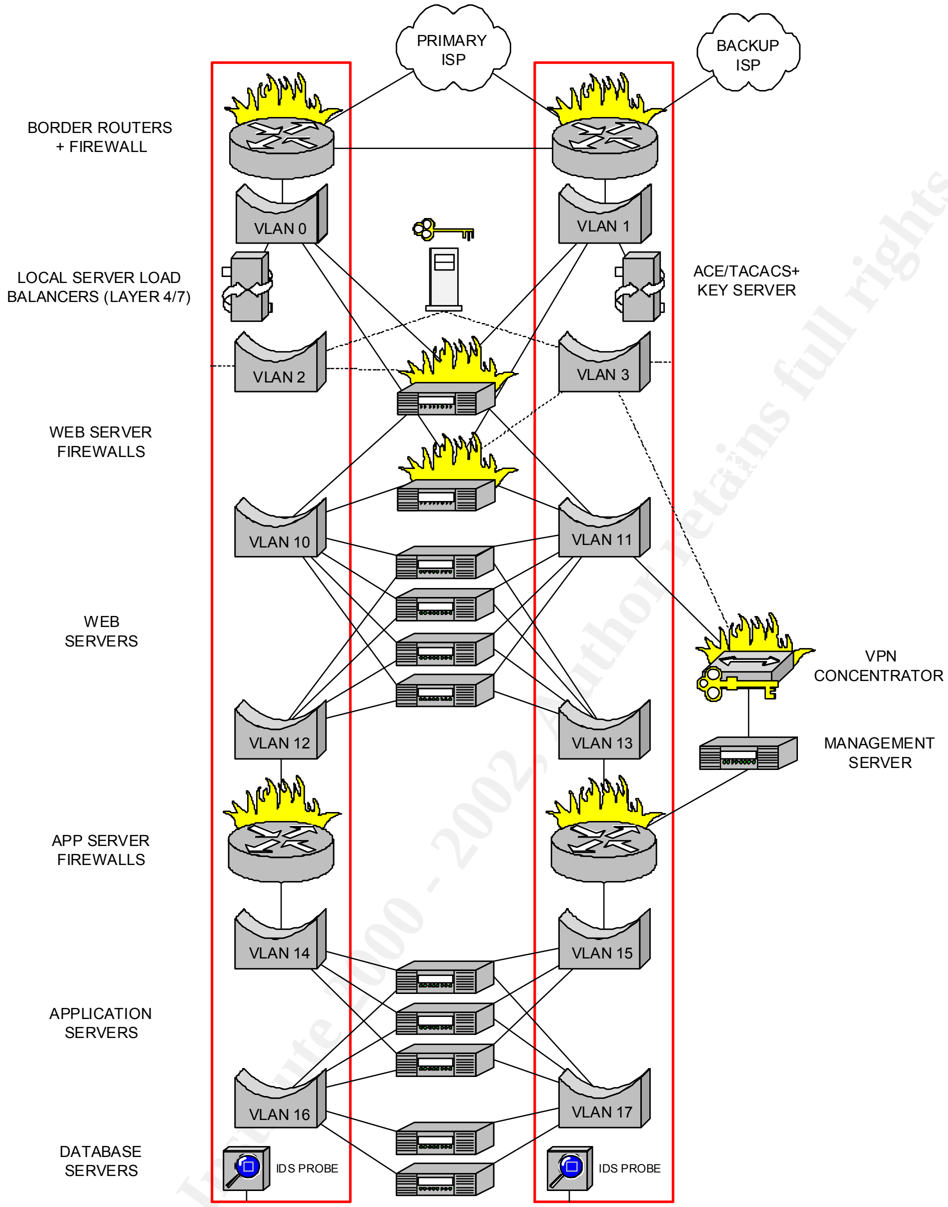
One switch is used for the even-numbered VLANs (the "left side") and another for the odd-numbered VLANs (the "right side"). Thus, if one switch fails, the network remains operational. Additionally, a high-end multifunction switch is used that support a number of other features within a single chassis, such as routing, firewalling, QoS, load balancing, and NIDS. For example, with the Cisco Catalyst 6509 core switch:

1. A multilayer switch feature card (MSFC) adds layer 3 routing capabilities, including packet filter firewalling, stateful packet inspection (reflection and content based access control), and BGP4.

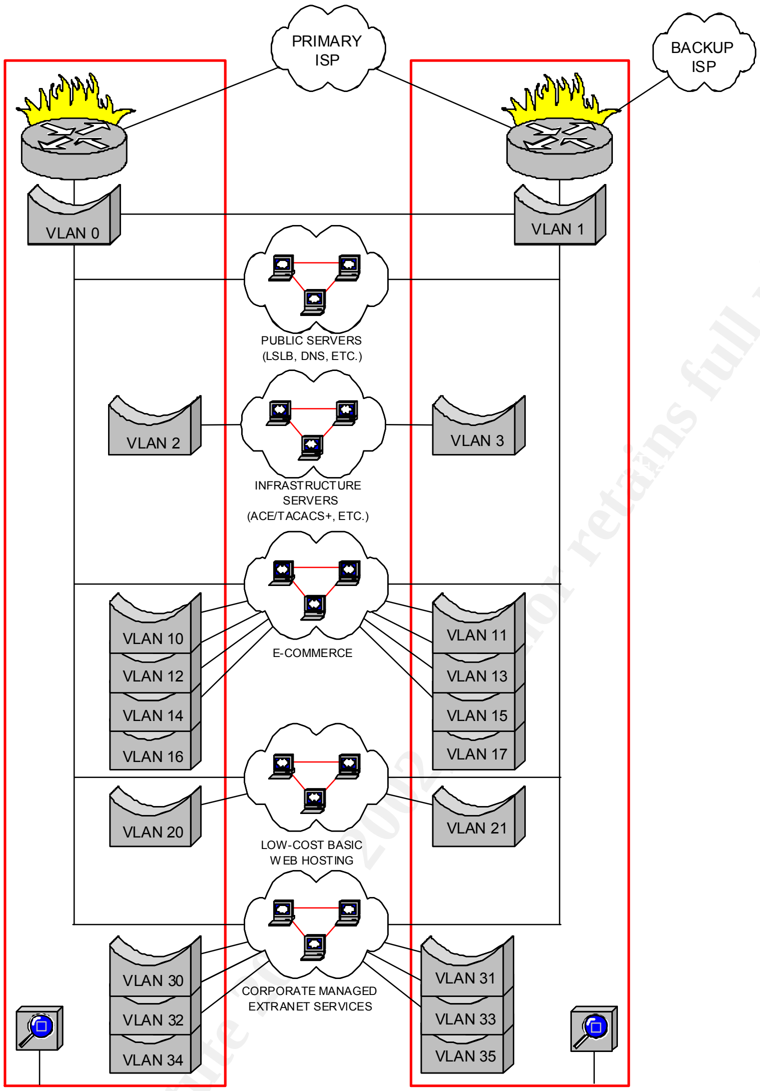
2. A policy feature card (PFC) adds advanced QoS.

3. A content switching module (CSM) adds layer 4/7 local server load balancing (similar to an ArrowPoint or Local Director).

4. An intrusion detection system blade (IDS) adds NIDS monitoring across *all* VLAN segments within the switch.



One box does it all. Better yet, the ASP can reuse the two switches to implement other products and services, such as low-cost basic web hosting, managed corporate extranets (pre-installed with groupware apps for e-mail, calendar, contacts, …), etc. Each new product is implemented using additional VLANs. Essentially, the two switches constitute the ASP's network infrastructure.



Each service network can either install a separate, dedicated firewall (such as the e-commerce service, which uses a PIX firewall between VLAN 0 and VLAN 10, as shown in §2.2.9 above, or use the built-in IOS firewalling capabilities of the router (for example, internal firewalling between VLAN 0 and VLAN 20).

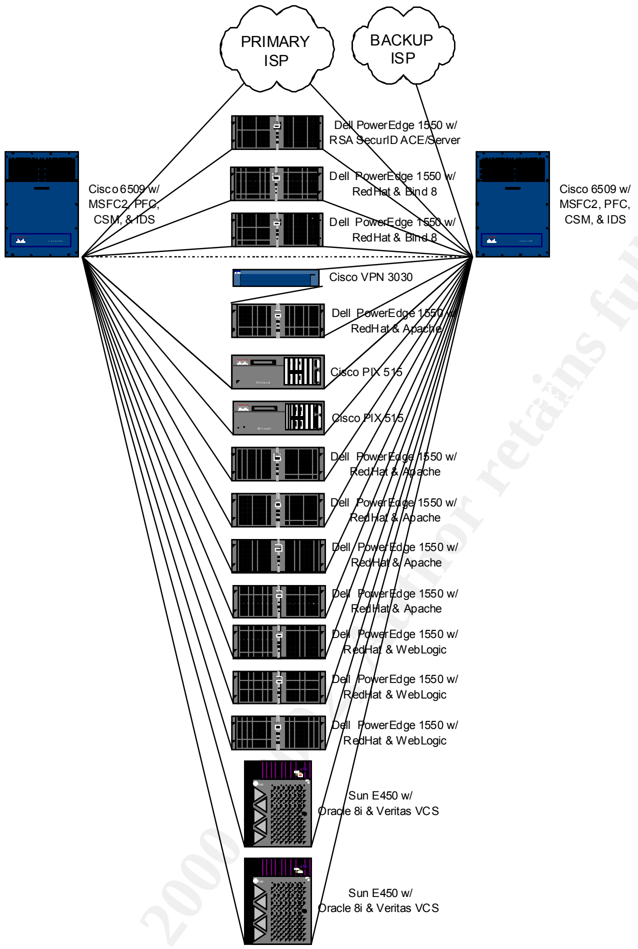
Now everyone is happy, including the CFO!

2.2.10 Physical Wiring

The physical wiring is deceptively simple. Essentially, almost everything plugs into both the two Cisco Catalyst 6509 switches. With exception of the web server firewalls[10] and VPN, the two 6509 switches perform all network functions.

[10] In fact, the firewall could have been added to the Catalyst 6509 in IOS on the MSFC card. However, the ASP felt it was wiser to use separate firewall devices and not to build *everything* into the switches.

The network looks deceptively simple when viewed this way.


PRIMARY ISP
BACKUP ISP
Cisco 6509 w/ MSFC2, PFC, CSM, & IDS
Cisco 6509 w/ MSFC2, PFC, CSM, & IDS
Dell PowerEdge 1550 w/ RSA SecurID ACE/Server
Dell PowerEdge 1550 w/ RedHat & Bind 8
Dell PowerEdge 1550 w/ RedHat & Bind 8
Cisco VPN 3030
Dell PowerEdge 1550 w/ RedHat & Apache
Cisco PIX 515
Cisco PIX 515
Dell PowerEdge 1550 w/ RedHat & Apache
Dell PowerEdge 1550 w/ RedHat & Apache
Dell PowerEdge 1550 w/ RedHat & Apache
Dell PowerEdge 1550 w/ RedHat & Apache
Dell PowerEdge 1550 w/ RedHat & WebLogic
Dell PowerEdge 1550 w/ RedHat & WebLogic
Dell PowerEdge 1550 w/ RedHat & WebLogic
Sun E450 w/ Oracle 8i & Veritas VCS
Sun E450 w/ Oracle 8i & Veritas VCS

2.3 RISKS

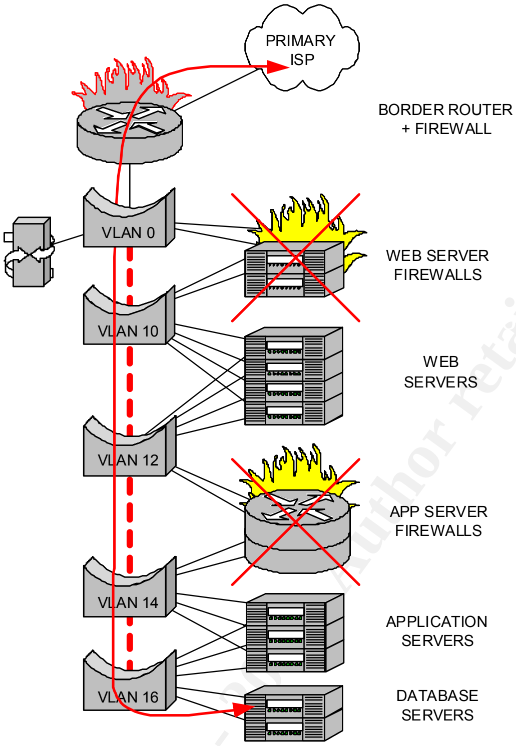
Using VLANs may lower costs and reduce the number of network components required, but these savings aren't without their risks.

- The network design presented in §2.2.7 above is complex. It will difficult to properly install, configure, and test, and even more difficult to keep properly configured over time. A single misconfiguration or bug in the switch's operating system (Catalyst and IOS) may open up the device to be hacked.

- Implementing switching, routing, firewalling, advanced QoS, local server load balancing, and NIDS within the same box is complex. In my experience, these devices become significantly less reliable (i.e., buggy) when too many features are used at the same time[11].

[11] Older versions of IOS broke by simply enabling both TCP/IP and NetBEUI. This intermittent bug took us *weeks* to diagnose!

- Implementing VLANs further increases complexity and risk. If the switch is compromised, then a hacker could potentially shunt the VLANs together to create a single, flat network and bypass all firewalls and cascaded network security!



For these reasons, I personally advise against using VLANs for network security. Unfortunately, I have encountered many facilities (and argued with many network engineers) that believe a properly configured VLAN is just a secure as using multiple independent, isolated switches.

Bottom line: As companies further cut their IT budgets to reduce overhead, it is increasingly difficult to get additional hardware approved for implementing better network security against what is deemed an unlikely, "theoretical" risk.

2.4 ADDITIONAL SECURITY IMPROVEMENTS

The following recommendations could be implemented to further improve security of the network. However, these recommendations come at a price, either in hard dollars or decreased network performance.

- Don't integrate the border router and core switch into the same device. Instead, use a separate dedicated router that is attached to the switch through a firewall. That way, even if the router's security is compromised, a hacker can not bypass the firewall by shunting VLANs together.

- Don't use VLANs for security! Rather, purchase independent switches to isolate security-sensitive network segments.

Unfortunately, this may significantly increase capital equipment costs.

- Add a packet filtering router between the VPN concentrator and management server, to filter out all traffic except TCP to ports 80 (http) and 443 (https) of the management server. This protects against a malicious vendor from attempting to hack into the management server.

  This can be implemented without purchasing any additional equipment by simply creating yet another VLAN with internal routing and filtering within the Catalyst 6509. However, this introduces additional complexity in the configuration.

- Use IPSEC for all internal communications between servers. That way, even if VLANs are hacked and shunted together, it is still difficult to directly access back-end systems, such as the database server.

  Unfortunately, this places additional load on the CPU to encrypt & decrypt packets. Crypto-accelerated NIC cards may help alleviate this.

- Instead of using a traditional layer-3 firewall (such as PIX or Firewall-1) for the application servers, a finer-grained application-level proxy may be used. This proxy would support the specific protocols used between the web server and the application server, such as RMI-IIOP, XML, or SOAP[12]. Firewall rules specify which specific *function calls* within the protocol are accepted or denied, thus providing finer-grained control over simple layer-4 filtering (e.g., port numbers).

  Unfortunately, these proxies are not very common, are typically vendor-specific (i.e., not cross-compatible), and may contain security vulnerabilities themselves, especially if the code used to implement them is reused from the application server!

- Global server load balancing (GSLB) may be used to create a second site in a different geographic region, thus providing security against local disasters (such as earthquakes and terrorism).

2.5 MISSING DETAILS

Admittedly, I culled a number of details in describing the network design, such as

- How do the application servers authorize credit cards for financial transactions? This implementation prevents them from directly accessing public online services like Verisign and eCash.

- How does the ASP monitor and manage the network? Wouldn't this require a management network with access to each of the VLANs?

[12] For product examples, refer to Orbix's Wonderwall or Fujitsu's Interstage IIOP gateway.

- How are bulk e-mail advertisements sent using the customer database?

The ASP will have other mechanisms built into the network, such as a management network for remotely monitoring and administrating the systems, a separate SAN (storage area network) or NAS (network attached storage) network for centralized disk storage and management, etc. Each of these mechanisms has its own unique design issues and must be carefully integrated into the network to avoid weakening security. However, this is well outside the scope of this humble paper and left as an exercise to the reader. ☺

3 SECURITY POLICIES

This section describes the security settings, VPN, and firewall rules for the:

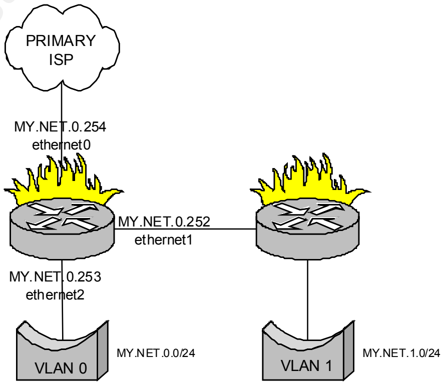
- Border routers / firewalls
- Web server firewalls
- Application server firewalls
- Management server VPN concentrator

In these policies, the ASP is assumed to own the public class B network MY.NET.0.0/16. In IOS command examples, the words "MY" and "NET" must be substituted with the actual octet values of the ASP's public class B address.

3.1 BORDER ROUTERS

In addition to connecting the ASP's autonomous network to the Internet, the two border routers are a central junction point for each of the ASP's products & services: e-commerce, basic web hosting, corporate managed extranets, etc.

This section illustrates the security policies for the left border router. The right border router is configured similarly.



3.1.1 E-Commerce Network ACLs

The e-commerce network (used by GIAC Enterprises to sell & distribute their online fortunes) has the strictest security policies of the ASP’s three services: e-commerce, basic web hosting, and corporate managed extranets.

Only http and https traffic is allowed to the e-commerce web servers. These servers do not publicly support any other protocols, such as ftp, telnet, smtp, pop, etc. The e-commerce systems are managed from a separate management web server, which is accessed through a secure VPN tunnel.

The border router implements this security policy through the following access control lists, applied to the Internet interface (ethernet 0):

1. Explicitly block all incoming traffic originating from illegal (or undesirable) IP addresses, defined as private IP addresses (per RFC 1918: 10.0.0.0/8, 172.16.0.0/9, and 192.168.0.0/16), local loopback (127.0.0.0/8), multicast (class D: 224.0.0.0/4), experimental (class E: 240.0.0.0/4), any MY.NET.0.0/16.

   ```
   ip access-list simple inet-illegal-addr-in
     remark “Drop private, loopback, class D-E, and my IP addresses”
     deny 10.0.0.0 0.255.255.255 log
     deny 127.0.0.0 0.255.255.255 log
     deny 172.16.0.0 0.15.255.255 log
     deny 192.168.0.0 0.0.255.255 log
     deny 224.0.0.0 31.255.255.255 log
     deny MY.NET.0.0 0.0.255.255 log
   ```

2. Only allow outgoing traffic from addresses within MY.NET.0.0/16. Implicitly block all other (spoofed) traffic.

   ```
   ip access-list extended inet-out
     remark “Drop spoofed packets”
     permit ip MY.NET.0.0 0.0.255.255 any
   ```

3. Allow BGP from neighbor exterior routers.

   ```
   ip access-list extended inet-router-in
     remark “Allow BGP to border router”

     !! My router = MY.NET.0.254
     !! Neighbor routers = THEIR.NET.0-1.254
     permit tcp THEIR.NET.0.254 0.0.1.0 host MY.NET.0.254 eq 179
   ```

4. Allow access to the “public servers” network: LSLB servers (for load balancing and redirection to web servers) and DNS servers.

   ```
   ip access-list extended inet-public-in
     remark “Allow access to public servers network”

     !! Allow HTTP & HTTPS to LSLB (MY.NET.0-1.1-127)
     !! (but not to the router @ MY.NET.0-1.1)
   ```

```
  deny tcp any MY.NET.0.0 0.0.1.0
  permit tcp any MY.NET.0.0 0.0.1.127 eq 80
  permit tcp any MY.NET.0.0 0.0.1.127 eq 443

  !! DNS servers = MY.NET.0-1.248-249, allow UDP only (not TCP)
  permit udp any MY.NET.0.248 0.0.1.1 eq 53
```

5. Allow access to the e-commerce network: web servers and management VPN.

```
ip access-list extended inet-ecommerce-in
  remark "Allow access to e-commerce servers"

  !! Allow HTTP & HTTPS to web servers (MY.NET.10-11.1-127)
  permit tcp any MY.NET.10.0 0.0.1.127 eq 80
  permit tcp any MY.NET.10.0 0.0.1.127 eq 443

  !! Allow IPSEC IKE & ESP to the management VPN (MY.NET.11.200)
  !! (IKE = ISAKMP = UDP port 500)
  !! (IPSEC ESP = IP protocol 50)
  permit udp any host MY.NET.11.200 eq 500
  permit 50 any host MY.NET.11.200
```

6. Apply the access control lists to the external interface (ethernet 0), thereby implicitly blocking any traffic that is not explicitly permitted in the lists.

```
interface ethernet 0
  remark "Internet"
  ip access-group inet-illegal-addr-in in
  ip access-group inet-router-in in
  ip access-group inet-public-in in
  ip access-group inet-ecommerce-in in
  ip access-group inet-out out
```

3.1.2 Web Hosting Network ACLs

The ASP's basic web hosting network is the most flexible and therefore implements significantly looser (weaker) security than the e-commerce network. Several protocols are supported from the Internet to these web servers, including http, https, ftp, smtp, pop, imap, telnet, ssh, and FrontPage server extensions. Furthermore, communications may be initiated from the web servers (such as DNS lookups from CGI scripts), so the firewall needs to allow for the resulting incoming traffic.

The border router implements these security policies through the following access control lists (in addition to the ACLs from §3.1.1 above), applied to the Internet interface (ethernet 0):

1. Allow basic access to the web servers.

```
ip access-list extended inet-basicweb-in
  remark "Allow access to basic web hosting servers"

  !! Allow access to web servers (MY.NET.20-21.1-127)
  !! (ftp=21, ssh=22, telnet=23, smtp=25, http=80, pop2=109,
  !! pop3=110, imap2=143, imap3=220, https=443)
  permit tcp any MY.NET.20.0 0.0.1.127 range 21 23
  permit tcp any MY.NET.20.0 0.0.1.127 eq 25
  permit tcp any MY.NET.20.0 0.0.1.127 eq 80
  permit tcp any MY.NET.20.0 0.0.1.127 eq 110
  permit tcp any MY.NET.20.0 0.0.1.127 eq 220
```

2. Allow reflexive outgoing traffic (from the web servers) that dynamically allows the returning incoming traffic:

```
  !! (This line is added to the access-list "inet-basicweb-in")
  !! Allow return traffic (from a reflexive rule) back in
  evaluate basicweb-reflect

ip access-list extended inet-basicweb-out
  remark "Allow reflexive outgoing traffic from basicweb servers"
!! Allow reflexive traffic out from web servers
  permit ip MY.NET.20.0 0.0.1.127 any reflect basicweb-reflect
```

3. Add the access lists "inet-basicweb-in" and "inet-basicweb-out" to the interface "ethernet 0", essentially making it:

```
interface ethernet 0
  remark "Internet"
  ip access-group inet-illegal-addr-in in
  ip access-group inet-router-in in
  ip access-group inet-public-in in
  ip access-group inet-ecommerce-in in
  ip access-group inet-basicweb-in in
  ip access-group inet-basicweb-out out
  ip access-group inet-out out
```

While a CBAC rule (content based access control) can be added to enable outgoing FTP by dynamically opening the data port, the performance impact to the router is immense. Therefore, the ASP advises customers to use passive FTP instead, at least until they have time to measure the potential performance impact.

If they supported the CBAC rule, it would look like:

```
ip inspect name ftp-cbac ftp

interface ethernet 0
  ip inspect ftp-cbac out
```

3.1.3 Corporate Managed Extranet ACLs

The corporate managed extranets include two components: a secure web port using only ports 80 and 443, and a VPN used to for LAN-to-LAN connection of an extranet to a customer's internal intranet. As such, the access control lists for the extranet network are nearly identical to those for the e-commerce network.

3.1.4 AAA Security Services

Login security of the Cisco 6509 is paramount. With administrative access, a hacker can reconfigure the device, disable firewalling (or worse, open up an obfuscated hole into the network), and shunt the VLANs.

Therefore, all login access is authenticated using Cisco’s AAA (authentication, authorization, and accounting) services. AAA is configured to authentication from the ACE/TACACS+ server, which in turn requires two-factor authentication from a SecurID crypto card with PIN pad.

Event logging is also sent to the TACACS+ server (similar to syslog).

Sidenote: The ACE server supports both RADIUS and TACACS+ protocols. However, Cisco’s AAA system does not fully support all AAA logging features to RADIUS (i.e., system and command events). Therefore, TACACS+ was selected.

The following commands are described in the AAA chapter of Cisco’s IOS Security Configuration Guide [3].

```
!! Specify the TACACS+ server and shared key.
tacacs-server host 10.0.2.1
tacacs-server key <password>

!! Enable AAA security services.
!! (AAA = Authentication, Authorization, and Accounting)
aaa new-model

!! Authenticate all logins and enable (privileged) access using
!! the TACACS+ server.  If TACACS+ is offline, then use the
!! enable password.  However, do NOT use the local user database!
aaa authentication login default group tacacs+ enable
aaa authentication enable default group tacacs+ enable

!! Log all accounting events to the TACACS+ server.
aaa accounting system default start-stop group tacacs+
aaa accounting network default start-stop group tacacs+
aaa accounting exec default start-stop group tacacs+
aaa accounting connection default start-stop group tacacs+
aaa accounting commands <n> default start-stop group tacacs+

!! Warning banners and error messages.
aaa authentication banner “Authorized access only, all activity
is logged”
aaa authentication fail-message “Login attempt failed and logged”
```

3.1.5 Routing Protocols

When using routing protocols, always require MD5 authentication. Beware against using *cleartext* authentication, which can be easily sniffed and defeated.

To use MD5 authentication with BGP, use:

```
router bgp <n>
  neighbor 145.2.2.2 password v61ne0qkel33&
```

```
To use MD5 authentication with OSPF, use:
```

```
ip ospf authentication-key <password>
ip ospf authentication message-digest
```

3.1.6 Misc. Device Hardening

The following additional security settings are recommended by the Cisco IOS Firewall Overview [2], SANS training module 2.3.1 [8], and yours truly.

```
!! Disable unnecessary/unused services
no ip bootp
no ip http
no service dhcp
no service finger
no service tcp-small-servers
no service udp-small-servers
no cdp run
no snmp
ntp disable

!! Disable source routing (very dangerous!)
no ip source-route

!! Disable directed broadcasts
no ip directed-broadcast

!! Do *not* disable ICMP "host unreachable" and
!! "protocol unreachable" messages.  These are necessary for
!! ICMP Path MTU discovery and can disrupt IPSEC if not enabled.
!! (see RFC 1195 and 1435)
!! Uncomment these lines (to disable ip unreachables) if IPSEC
!! is not used.
!!no ip unreachables

!! Disable ICMP redirects
no ip redirects

!! Disable ICMP Router Discovery Protocol (IRDP), unless needed
no ip irdp

!! Disable ICMP mask reply (to query for netmask)
no ip mask-reply

!! Disable proxy ARP
interface ethernet 0
  no ip proxy-arp

!! Encrypt passwords (albeit weak, crackable encryption)
service password encryption

!! Enable syslog logging
```

```
logging 10.0.2.2
logging trap 6
```

ICMP unreachable messages are left enabled per to Cisco's recommendation for compatibility with IPSEC:

> *"We recommend 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."* [12]

Finally, when setting the administrative password (as a backup to AAA authentication when the TACACS+ server is down), use **`enable secret`** instead of **`enable password`**, since it implements stronger password encryption.

3.1.7 Additional Security Improvements

For added security, TCP Intercept can be configured on the border router to protect against SYN floods. The router intercepts SYN packets and performs three-way handshakes on behalf of the destination servers. If/when a handshake completes, the router "forwards" the connection on to the server.

However, with the current configuration, this is unnecessary. All TCP-based servers are currently protected, as follows:

- The PIX firewalls (see §3.2 below) are configured to protect the e-commerce web servers against SYN floods (using its "Embryonic Connection Limit").
- The LSLB devices implicitly perform a similar form of TCP SYN interception.
- The DNS servers only receiving UDP (TCP is filtered out) and therefore are not vulnerable to SYN floods.

If a public TCP-based server is installed onto an unprotected network, such as VLAN 0 or VLAN 1, then the border router can be configured to protect it by the following commands:

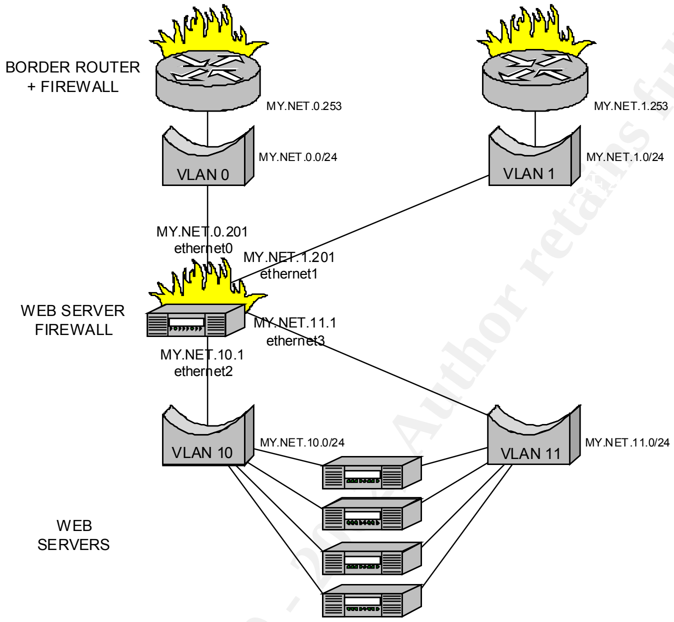
```
ip access-list <num> permit tcp any host <host-ip> [<ports>]
ip tcp intercept list <num>
ip tcp intercept mode intercept
```

Note: The "tcp intercept" command requires an extended ACL that is referenced by number (100-199) rather than by name. Hopefully, a future version of IOS will add support for named access lists.

3.2 WEB SERVER FIREWALLS

The web server firewalls (for the e-commerce network) perform a number of additional security protections beyond those implemented on the border router firewall, such as protections against SYN flood and fragmented packet attacks. They also protects against attacks launched from within the ASP's other service networks, such as basic web hosting network or corporate managed extranets, should they become compromised.

This section illustrates security policies for the left web server firewall. The right firewall is configured similarly.



3.2.1 Access Control Lists

The following access control lists restrict access to the web servers to http and https traffic only. They also open a hole to allow IPSEC traffic (IKE and ESP) from the Internet to the management server's VPN concentrator.

Unlike IOS, which allows multiple ACLs to applied to an interface by using multiple "access-group" commands, the PIX system only allows a single ACL list for each direction of an interface (one list for the "in" direction, and one for the "out").

1. Configure the network interfaces. By setting VLANs 10-11 at a higher security level than VLANS 0-1 (security100 vs. security0, respectively), outgoing traffic (from VLANs 10-11 to VLANs 0-1) is implicitly allowed while incoming traffic (from VLANs 0-1 to VLANs 10-11) is implicitly blocked.

   Prevent any traffic to the TACACS+ network (except for AAA authentication) on VLAN2 by setting it to the highest security level (security255).

   ```
   nameif ethernet0 vlan0 security0
   nameif ethernet1 vlan1 security0
   nameif ethernet4 vlan2 security255
   nameif ethernet2 vlan10 security100
   nameif ethernet3 vlan11 security100
   ```

```
ip address vlan0 MY.NET.0.201 255.255.255.0
ip address vlan1 MY.NET.1.201 255.255.255.0
ip address vlan2 10.0.2.201 255.255.255.0
ip address vlan10 MY.NET.10.1 255.255.255.0
ip address vlan11 MY.NET.11.1 255.255.255.0
```

2. Explicitly block all incoming traffic originating from illegal (or undesirable) IP addresses, defined as private IP addresses (per RFC 1918: 10.0.0.0/8, 172.16.0.0/9, and 192.168.0.0/16), local loopback (127.0.0.0/8), multicast (class D: 224.0.0.0/4), and experimental (class E: 240.0.0.0/4).

   ```
   access-list ingress deny ip 10.0.0.0 0.255.255.255 any
   access-list ingress deny ip 127.0.0.0 0.255.255.255 any
   access-list ingress deny ip 172.16.0.0 0.15.255.255 any
   access-list ingress deny ip 192.168.0.0 0.0.255.255 log
   access-list ingress deny ip 224.0.0.0 31.255.255.255 any
   ```

3. Allow access to the e-commerce network: web servers and management VPN.

   ```
   !! Allow HTTP & HTTPS to web servers (MY.NET.10-11.1-127)
   access-list ingress permit tcp any MY.NET.10.0 0.0.1.255 eq 80
   access-list ingress permit tcp any MY.NET.10.0 0.0.1.255 eq 443

   !! Allow IPSEC IKE & ESP to the management VPN (MY.NET.11.200)
   !! (IKE = ISAKMP = UDP port 500)
   !! (IPSEC ESP = IP protocol 50)
   access-list ingress permit udp any host MY.NET.11.200 eq 500
   access-list ingress permit 50 any host MY.NET.11.200
   ```

4. Only allow outgoing traffic from addresses within MY.NET.10-11.0/24. Explicitly block all other (spoofed) traffic. (Note: This rule is more restrictive than the analogous egress rule on the border router.)

   ```
   access-list egress permit tcp MY.NET.10.0 0.0.1.255 eq 80 any
   access-list egress permit tcp MY.NET.10.0 0.0.1.255 eq 443 any
   access-list egress deny ip any any
   ```

5. Apply the access control lists to the external interfaces: vlan0 and vlan1.

   ```
   access-group ingress in interface vlan0
   access-group ingress in interface vlan1

   access-group egress out interface vlan0
   access-group egress out interface vlan1
   ```

3.2.2 Embryonic Connection Limit

For added security, TCP Intercept is configured on the PIX firewall to protect the web servers against SYN floods. The firewall intercepts SYN packets and performs three-way handshakes on behalf of the destination servers. If/when a handshake completes, the router "forwards" the connection on to the server.

The PIX Firewall Command Reference describes this feature as follows:

> *"...once the optional embryonic connection[13] limit is reached, and until the embryonic connection count falls below this threshold, every SYN bound for the affected server is intercepted. For each SYN, PIX Firewall responds on behalf of the server with an empty SYN/ACK segment. PIX Firewall retains pertinent state information, drops the packet, and waits for the client's acknowledgement. If the ACK is received, then a copy of the client's SYN segment is sent to the server and the TCP three-way handshake is performed between PIX Firewall and the server. If and only if, this three-way handshake completes, may the connection resume as normal. If the client does not respond during any part of the connection phase, then PIX Firewall retransmits the necessary segment using exponential back-offs."* [13]

TCP Intercept (a.k.a.. "Flood Guard", a.k.a., "Embryonic Connection Limit") is enabled as follows:

```
!! Set the embryonic connection limit to 100
static (vlan10,vlan0) MY.NET.10.0 MY.NET.10.0 \
    netmask 255.255.255.0 5000 100
static (vlan11,vlan0) MY.NET.10.0 MY.NET.10.0 \
    netmask 255.255.255.0 5000 100
static (vlan10,vlan1) MY.NET.10.0 MY.NET.10.0 \
    netmask 255.255.255.0 5000 100
static (vlan11,vlan1) MY.NET.10.0 MY.NET.10.0 \
    netmask 255.255.255.0 5000 100
```

3.2.3 AAA Security Services

All console login access is authenticated using Cisco's AAA (authentication, authorization, and accounting) services. AAA is configured to authentication from the ACE/TACACS+ server, which in turn requires two-factor authentication from a SecurID crypto card with PIN pad.

Event logging is also sent to the TACACS+ server (similar to syslog).

```
!! Specify the TACACS+ server and shared key.
aaa-server TACACS+ (vlan2) host 10.0.2.1 <password> timeout 20

!! Authenticate all console access using the TACACS+ server.
aaa authentication console TACACS+

!! Log all accounting events to the TACACS+ server.
aaa accounting all vlan0 TACACS+
aaa accounting all vlan1 TACACS+
aaa accounting all vlan2 TACACS+
aaa accounting all vlan10 TACACS+
aaa accounting all vlan11 TACACS+
```

[13] An "*embryonic connection*" is just a fancy name for an uncompleted TCP three-way handshake.

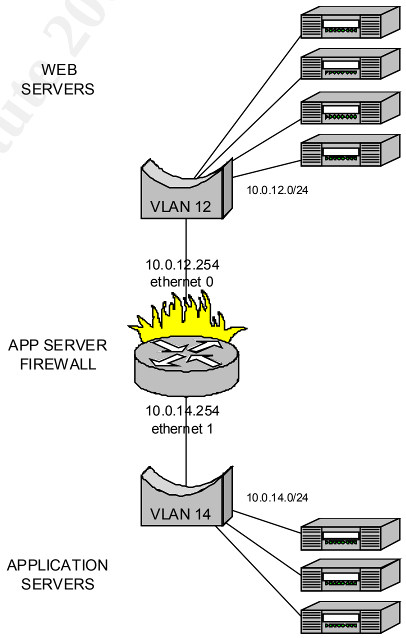
3.2.4 Misc. Device Hardening

The following additional security settings disable unnecessary services and enable logging to the syslog server.

```
!! Disable PIX administration via telnet or http (use ssh!)
no telnet
no http server enable

!! Disable unnecessary/unused services
no dhcpd enable

!! ICMP unreachable is necessary for ICMP Path MTU discovery
!! and can disrupt IPSEC if not enabled.  (see RFC 1195 and 1435)
!! Comment out these lines if IPSEC is not used.
icmp permit any unreachable vlan0
icmp permit any unreachable vlan1

!! Enable syslog logging
logging host 10.0.2.2
logging timestamp
logging trap 5
logging on
```

3.3 APPLICATION SERVER FIREWALLS

Configuration of the application server firewalls is quite simple. The web servers communicate to the application servers using the well-known port range 1000-1010. The web servers always initiate communication.



A reflexive rule could be applied to deny access from the app servers except when first initiated by the web server. However in practice, this is an unnecessary complication: if the app server (or the app server's VLAN) is compromised, then the hacker implicitly has control over the web servers anyway.

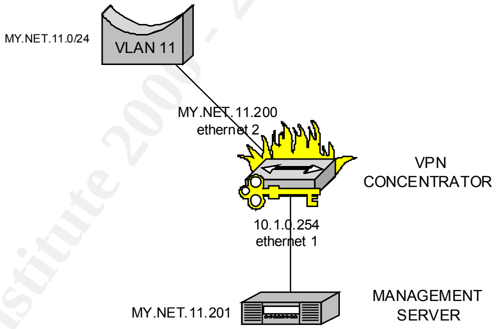
Therefore, only the following basic rules are necessary:

```
ip access-list extended ws-in
  remark "Allow web servers to app servers ports 1000-1010"
  permit tcp 10.0.12.0 0.0.0.255 10.0.14.0 0.0.0.255 range 1000 1010

interface ethernet 0
  ip access-group ws-in in
```

AAA security services and device hardening is configured globally per §3.1.2 and §3.1.6 above.

3.4 MANAGEMENT VPN

Unlike the Catalyst 6509 switch/router and the PIX firewall, which is configured using command scripts, the VPN 3030 concentrator is typically configured using a web-based GUI. Therefore, this section describes the security policies in general rather than giving the precise commands used to configure the device. Where appropriate, references are made to the specific web forms used to enter the policies along with a URL to online Cisco documentation for further assistance. However, for brevity, a detailed walk-through (including screen grabs of the web interface) is not included.



3.4.1 Security Policy

The VPN concentrator is configured to create a secure tunnel between vendors and the management server[14] used to manage and administer their e-commerce site. The 3030 concentrator was specifically chosen due to its support for hardware encryption (50Mb/s encryption throughput) and up to 1500 simultaneous users, which well meets/surpasses the performance requirements for the management network.

[14] Remember, the management server is nothing more than a web server connected to the application servers that runs special software for managing the e-commerce sites, including user, content, and site administration. Access to this special web server is protected by a secure VPN tunnel in order to provide an extra layer of protection.

The IPSEC tunnel is configured to use the Cisco-supplied SA "ESP-3DES-MD5" exclusively, which is defined as follows (per [18]):

- ESP protocol with MD5/HMAC-128 authentication and 3DES-168 encryption.
- Tunneling encapsulation.
- Perfect forward secrecy disabled.
- IKE key exchange, phase 1 using main mode, and phase 2 using MD5/HMAC-128 for authentication, DES for encryption, and Diffie-Hellman Group 1 to generate the SA keys.

This provides sufficiently strong encryption *and* authentication for all VPN traffic while maintaining high performance of the VPN concentrator hardware.

Additionally, the VPN concentrator uses SecurID two-factor user authentication for all connections, performed through the ACE server.

Split tunneling is explicitly prohibited. Thus, whenever a vendor connects to the management server through the VPN, their PC should be automatically disconnected from any other network connections, including their LAN and the Internet. Since client PCs should never be able to connect to both the Internet *and* the VPN at the same time, it cannot be (as easily) exploited as a router from the Internet through the VPN.

3.4.2 Basic Configuration

In order to configure the basic security policies described in §3.4.1 above, perform the following steps:

1. From "**Configuration | User Management | Base Group Screen, General tab**" [16], set:

    - Tunneling Protocols = IPSEC only (disable all other protocols: PPTP, L2TP, and L2TP over IPSEC)

2. From "**Configuration | User Management | Base Group Screen, IPSEC Parameters tab**" [17], set:

    - IPSEC SA = ESP-3DES-MD5
    - Tunnel Type = Remote access (client-to-LAN only)
    - Authentication = SDI (the SecurID ACE server)
    - Mode Configuration = enabled
    - Banner = "Authorized Access Only"
    - Allow Password Storage on Client = disabled (evil!)
    - Split Tunneling Policy = disabled
    - Allow the Networks in List to Bypass Tunnel = disabled

3. From "**Configuration | User Management | Groups | Authentication Servers | Add or Modify SDI Screen**" [15], set:

    - Server Type = SDI (the SecurID ACE server)
    - Authentication Server = 10.0.2.1

4 SECURITY POLICY TUTORIAL

This section gives a basic tutorial for implementing the security rules for the border router, including a basic introduction to IOS firewall configuration commands.

For a line-by-line walkthrough of the configuration script, please refer to the detailed descriptions in §3 above.

4.1 CONFIGURATION SCRIPTS

In my experience, the best practice for configuring and maintaining Cisco gear is using offline configuration scripts and NOT interactive GUIs or command line interfaces (CLIs).

- All configurations should be implemented using ASCII-based script files. If the device fails or needs to be reset (clearing the NVRAM), then the configuration can be reloaded directly from the scripts.

- When updating the configuration of a device, the changes should be made to the scripts and then the scripts reapplied. This ensures that the scripts are accurate, error-free (including type-o's), and properly reflect the current configuration of the device.

- Configuration scripts should be stored into a source code management system, such as CVS (see §4.2 below), thus preserving a copy of all past versions of the scripts along with a textual "change log" describing what was changed for each new version, by whom, and why.

Cisco similarly recommends this practice, particularly with access control lists. Cisco's IOS Security Configuration Guide specifically states:

> *"Because the order of access list criteria statements is important, and because you cannot reorder or delete criteria statements on your router, Cisco recommends that you create all access list statements on a TFTP server, and then download the entire access list to your router.*
>
> *To use a TFTP server, create the access list statements using any text editor, and save the access list in ASCII format to a TFTP server that is accessible by your router. Then, from your router, use the* ***copy tftp:****file_id* ***system:running-config*** *command to copy the access list to your router. Finally, perform the* ***copy***

> ***system:running-config nvram:startup-config*** *command to save the access list to your router's NVRAM.*
>
> *Then, if you ever want to make changes to an access list, you can make them to the text file on the TFTP server, and copy the edited file to your router as before."* [11]

4.2 CVS

If nothing else, I recommend using CVS (Concurrent Version System) for source code management, including router configuration scripts.

- CVS maintains a copy of all past version of a file, including a history trail, textual description of the changes, optional approval/release status of the files, etc. CVS options allow you to display the differences between any two versions of a file, based on version number, file check-in date, or symbolic tag.

- Multiple users can access and modify files simultaneously within a CVS repository. If two users modify the same version of the file, they can elect to either (1) create a new branch, (2) have CVS automatically merge the two versions, if the changes do not overlap (3) manually merge the changes, or (4) override the previous version, or (5) discard changes. This decision doesn't need to be made until check-in time, and then it is only required when necessary to resolve a change conflict.

- Alternatively, files may be optionally locked by the user when checked-out, to ensure that others only have read-only access to them.

- CVS is SUPER-simple to setup, administer, backup, and restore. This is contrasted against expensive commercial source code management systems that patch into the operating system kernel and/or require administrators to be trained for several days to learn how to setup, administer, and (God forbid!) backup & restore their systems.

- CVS is robust! In my 10 years (as a programmer and director of engineering), it has never lost or corrupted a single version of any files stored in its repository. It is well used and tested by the open source community. Contrast this with other commercial systems that have a checkered past of repository corruption! Ugh!

- CVS is supported on multiple platforms (all flavors of Unix, plus Windows, NT, and Macintosh), supports both command-line and GUI "file browser style" interfaces, and is integrated into many commercial software packages.

- It supports a number of security methods, including Kerberos authentication.

- Finally, the price it right… FREE. But I'd use it even if it cost money.

More information on CVS is available at http://www.gnu.org/software/cvs/cvs.html.

Alternative source code management systems include:

- Rational ClearCase – A very advanced system, but also pricey and complex.
- RCS – A very simple tool intended primarily for single person use, but its free. In fact, CVS is built on top of RCS.
- Microsoft SourceSafe – Beware, I've seen this corrupt files in past versions!

I strongly advise storing all configuration files into a source code management system. It makes tracking changes so much easier, especially when troubleshooting failures resulting from configuration changes!

4.3 CONFIGURING THE BORDER ROUTER

The following steps are used to apply the security policy configurations from §3.1 above to the border router.

4.3.1 Step 1 – Create a Configuration Script File

Create an ASCII-based configuration script containing all of the relevant IOS commands. For example, to configure the left border router for both the e-commerce and basic web hosting networks (as described in §3.1.1 and §3.1.2 above), use the following script[15]:

```
!! left-router-security.ios – Configure security for border router MY.NET.0.254.
!!
!! Configures access for public servers, e-commerce network, and basic web
!! hosting network.  Also hardens the router and configures TACACS+ authentication
!! and syslog logging.
!!
!! $Header$
!! (c) ASP's Company Name, 2001 – All Rights Reserved


ip access-list simple inet-illegal-addr-in
  remark "Drop private, loopback, class D-E, and my IP addresses"
  deny 10.0.0.0 0.255.255.255 log
  deny 127.0.0.0 0.255.255.255 log
  deny 172.16.0.0 0.15.255.255 log
  deny 192.168.0.0 0.0.255.255 log
  deny 224.0.0.0 31.255.255.255 log
  deny MY.NET.0.0 0.0.255.255 log


ip access-list extended inet-out
  remark "Drop spoofed packets"
  permit ip MY.NET.0.0 0.0.255.255 any


ip access-list extended inet-router-in
  remark "Allow BGP to border router"

  !! My router = MY.NET.0.254
  !! Neighbor routers = THEIR.NET.0-1.254
  permit tcp THEIR.NET.0.254 0.0.1.0 host MY.NET.0.254 eq 179


ip access-list extended inet-public-in
```

[15] Don't forget that "MY.NET" must be replaces with the actual class B network address of the ASP in order for this script to execute without errors!

```
  remark “Allow access to public servers network”

  !! Allow HTTP & HTTPS to LSLB (MY.NET.0-1.1-127)
  !! (but not to the router @ MY.NET.0-1.1)
  deny tcp any MY.NET.0.0 0.0.1.0
  permit tcp any MY.NET.0.0 0.0.1.127 eq 80
  permit tcp any MY.NET.0.0 0.0.1.127 eq 443

  !! DNS servers = MY.NET.0-1.248-249, allow UDP only (not TCP)
  permit udp any MY.NET.0.248 0.0.1.1 eq 53


ip access-list extended inet-ecommerce-in
  remark “Allow access to e-commerce servers”

  !! Allow HTTP & HTTPS to web servers (MY.NET.10-11.1-127)
  permit tcp any MY.NET.10.0 0.0.1.127 eq 80
  permit tcp any MY.NET.10.0 0.0.1.127 eq 443

  !! Allow IPSEC IKE & ESP to the management VPN (MY.NET.11.200)
  !! (IKE = ISAKMP = UDP port 500)
  !! (IPSEC ESP = IP protocol 50)
  permit udp any host MY.NET.11.200 eq 500
  permit 50 any host MY.NET.11.200


ip access-list extended inet-basicweb-in
  remark “Allow access to basic web hosting servers”

  !! Allow access to web servers (MY.NET.20-21.1-127)
  !! (ftp=21, ssh=22, telnet=23, smtp=25, http=80, pop2=109,
  !! pop3=110, imap2=143, imap3=220, https=443)
  permit tcp any MY.NET.20.0 0.0.1.127 range 21 23
  permit tcp any MY.NET.20.0 0.0.1.127 eq 25
  permit tcp any MY.NET.20.0 0.0.1.127 eq 80
  permit tcp any MY.NET.20.0 0.0.1.127 eq 110
  permit tcp any MY.NET.20.0 0.0.1.127 eq 220

  !! Allow return traffic (from a reflexive rule) back in
  evaluate basicweb-reflect


ip access-list extended inet-basicweb-out
  remark “Allow reflexive outgoing traffic from basicweb servers”
!! Allow reflexive traffic out from web servers
  permit ip MY.NET.20.0 0.0.1.127 any reflect basicweb-reflect


interface ethernet 0
  remark “Internet”
  ip access-group inet-illegal-addr-in in
  ip access-group inet-router-in in
  ip access-group inet-public-in in
  ip access-group inet-ecommerce-in in
  ip access-group inet-basicweb-in in
  ip access-group inet-basicweb-out out
  ip access-group inet-out out
```

The first line of the script contains the RCS keyword “`$Header$`”. This keywords is automatically replaced with file version information whenever the file is checked in or out via CVS. To include a complete log of the file’s change history, add the line “`!! $Log$`” into the script.

4.3.2 Step 2 – Backup the Current Router's Configuration

Use the IOS "**copy**" command to backup the router's current running configuration. For example, to backup the current running configuration:

```
#copy run tftp
Address of name of remote host []? 10.0.2.100
Destination filename [router-config]? left-router-config-before
!!
1096 bytes copied in 5.27 secs (208 bytes/sec)
#
```

This way, if something goes terribly wrong, then you can restore the router's running configuration using the command "`copy tftp run`", *even if the running configuration is accidentally different than the startup configuration in NVRAM!*

4.3.3 Step 3 – Load the Configuration Script

Next, use "**copy**" to load the security policy configurations.

```
#copy tftp run
Address of name of remote host []? 10.0.2.100
Source filename [router-config]? left-router-security.ios
Accessing tftp://10.0.2.10/left-router-security.ios
Loading left-router-security.ios from 10.0.2.100 (via ethernet3)
1096 bytes copied in 5.27 secs (208 bytes/sec)
!!
[OK - 796/2048 bytes]
796 bytes copied in 8.43 secs (94 bytes/sec)
#
```

4.3.4 Step 4 – Backup the New Running Configuration

(Optional) Backup the new running configuration to file.

```
#copy run tftp
Address of name of remote host []? 10.0.2.100
Destination filename [router-config]? left-router-config-after
!!
1408 bytes copied in 3.78 secs (372 bytes/sec)
#
```

4.3.5 Step 5 – Test the Router

Check the configuration of the router by reviewing the new running configuration file (§4.3.4 above), using the IOS commands "**show**", and testing and auditing the router (§5 below).

4.3.6 Step 6 – Save the Configuration to NVRAM

If everything tests out ok, then it is necessary to save the running configuration into NVRAM. Otherwise, the router will resort to its old configuration the next time it is rebooted!

Use the command "**`copy run start`**" to save the "running-config" into the NVRAM file "startup-config".

4.3.7 Step 5 – Check-in the File to CVS

Execute the command "**`cvs commit left-router-security.ios`**" to store the file into CVS.

I would also suggest storing the "before" and "after" backups of the router's running configuration into CVS, just in case. These files can be extremely useful for troubleshooting network problems down the road. Furthermore, if an intrusion occurs, these files can help determine if the router was compromised or tampered, which is especially useful during the "Identification" phase of Incident Handling (see SANS Track 4: Incident Handling & Response).

That's all, folks! The border router is now configured with the security policies.

4.4 Quick Introduction to IOS Firewall Commands

The IOS configuration and firewalling commands used in the configuration script are documented in four online references [1]:

- Cisco IOS IP and IP Routing Configuration Guide, Release 12.1
- Cisco IOS IP and IP Routing Command Reference, Release 12.1
- Cisco IOS Security Configuration Guide, Release 12.1
- Cisco IOS Security Command Reference, Release 12.1

The primary commands used to create the access control lists are:

- **`ip access-list simple <name>`**

  Creates a new simple access control list, used to filter packets only using the source IP address. This is much faster then extended ACLs (below), albeit much less functional.

- **`ip access-list extended`**

  Creates a new extended access control list, used to filter packets by source & destination address, protocol, port number (for TCP and UDP), and message code & type (for ICMP).

Extended ACLs also support reflexive rules, allowing outgoing traffic to dynamically allow the return traffic back in.

- **`permit <rule>`**

  Adds a “permit” rule to the current access-list. If the rule criteria matches the current packet, then the packet is immediately allowed through.

  For example:

  ```
        permit tcp any host 1.2.3.4 eq 80
  ```

  enables any computer to access host 1.2.3.4 port 80 (typically a web server).

- **`deny <rule>`**

  Adds a “deny” rule to the current access-list. If the rule criteria matches the current packet, then the packet is immediately discarded.

  For example:

  deny 10.0.0.0 0.255.255.255 log

  causes any traffic with a source IP address within 10.0.0.0/ to be dropped. Furthermore, the “log” option causes this rule to be logged (to syslog, etc.) whenever it is executed.

- **`ip access-group`**

  Applies an access-list to the current interface.

  For example:

  ```
        interface ethernet 0
          ip access-group ingress1 in
          ip access-group ingress2 in
  ```

  Applies the two ACLs named “ingress1” and “ingress2”, in order, to all incoming packets to interface “ethernet 0”.

If an interface has no ACLs defined for it, then it allows all packets through. However, if one or more ACLs are defined, then the interface drops all packets that are not explicitly permitted by the ACLs.

Rule ordering is important. Rules are tested sequentially, and only the first rule that matches is executed. Therefore, if a packet matches two different rules, the first rule is applied while the second rule is silently ignored.

Be careful when specifying IP addresses in permit and deny rules. The general format is:

```
<ip-address>  <wildcard>
```

For example,

```
10.0.0.0  0.255.255.255
```

The second value is **not** a netmask but rather a wildcard bitmap, which is the opposite of a bitmap[16]. A wildcard bit value of “0” means corresponding ip-address bit value must match the packet’s address. A wildcard bit “1” means the corresponding ip-address bit is ignored and does not have to match.

In other words, to determine if a packet’s address “P” matches a rule’s address “R” and wildcard “W”, use the following expression (using ISO C notation, evaluates to 0 if false and non-zero if true):

(P & ~W) == (R & ~W)

or more efficiently:

(P ^ R) & ~W

Please note how wildcards are used within the configuration script to specify more than simple netmaps. For example, the following permit rule from the ACL inet-public-in uses the address “MY.NET.0.248” with a wildcard of “0.0.1.1” to specify the address range MY.NET.0-1.248-249. This type of wildcarding is not possible using normal netmasks.

```
!! DNS servers = MY.NET.0-1.248-249, allow UDP only (not TCP)
permit udp any MY.NET.0.248 0.0.1.1 eq 53
```

4.5 Example – Testing ACL Rules

This section gives a brief example on how to test the following five ACL rules to ensure they are applied and working properly:

```
ip access-list simple inet-illegal-addr-in
  remark “Drop private, loopback, class D-E, & my IP addresses”
  deny 10.0.0.0 0.255.255.255 log
  deny 127.0.0.0 0.255.255.255 log
  deny 172.16.0.0 0.15.255.255 log
  deny 192.168.0.0 0.0.255.255 log
  deny 224.0.0.0 31.255.255.255 log
  deny MY.NET.0.0 0.0.255.255 log
```

The commands “`show access-lists`” and “`show ip access-list`” may be used to display the contents of an ACL. For example:

```
#show access-lists inet-illegal-addr-in

Extended IP access list inet-illegal-addr-in
  deny 10.0.0.0 0.255.255.255 log
  deny 127.0.0.0 0.255.255.255 log
  deny 172.16.0.0 0.15.255.255 log
  deny 192.168.0.0 0.0.255.255 log
  deny 224.0.0.0 31.255.255.255 log
```

[16] Some people call the wildcard an “inverse netmask”, since it is the ones-complement of a netmask. While this is true, it is also much more versatile than a bitmask, as used within our configuration scripts.

The commands "**show ip interfaces**" and "**show running-config**" may be used to display which interfaces are configured with ACLs.

The configuration as a whole for the router can be determined by inspecting the running-config backup file (§4.3.4 above). I thoroughly recommend glacing though this file.

Finally, perhaps the best test is to inject some real network traffic and observe the results. Connect a laptop containing Nmap to the ethernet0 interface (perhaps using a hub so-as not to disconnect the primary connection), another laptop running tcpdump to ethernet1, and attempt to send packets to ethernet1 using port scans with spoofed source IP addresses (10.0.0.0/8, 172.16.0.0/9, etc.).

Obviously, the Nmap scans should fail. If they succeed, then you know something is seriously wrong and your firewall is broken.

Even if Nmap fails (as it should), perform the following additional checks:

- The command "**show access-lists**" displays if any rules succeeded and how many times. Check these numbers against the number of illegal packets sent, to make sure they were all properly found and rejected.

  For example:

  ```
  #show access-lists inet-illegal-addr-in

  Extended IP access list inet-illegal-addr-in
    deny 10.0.0.0 0.255.255.255 log  (197 matches)
    deny 127.0.0.0 0.255.255.255 log
    deny 172.16.0.0 0.15.255.255 log  (57 matches)
    deny 192.168.0.0 0.0.255.255 log  (65535 matches)
    deny 224.0.0.0 31.255.255.255 log
  ```

- Inspect the syslog to determine if packets are being properly dropped. Note: The syslog is not guaranteed to log an entry for every packet that is dropped. If too many syslog entries are being output at the same time, then some of the entries will be silently dropped. Otherwise, the router can become quickly overloaded with syslog messages, essentially opening the door to a DoS attack!

- Finally, even if everything else reports that the packets are being filtered correctly, check the tcpdump sniffer as a sanity check to ensure they aren't sneaking through, somehow.

5 AUDITING THE SECURITY ARCHITECTURE

This section develops a basic test plan for validating the network and auditing configurations and access control rules.

This section is not intended to design a full-scale "*enterprise audit*"[17] and risk assessment. These typically start by identifying critical assets (including information assets), business processes, and asset flows through those processes, determining business risks and potential damages, performing perimeter analysis, identifying vulnerabilities, and finally recommending security measures to mitigate significant risks.

Rather, this section presents the humble beginnings of a "*configuration validation*" audit, similar to QA testing, used to verify that predefined security policies are configured and operating correctly on the individual network components.

5.1 Test Plan

Auditing the border router involves the following basic steps:

1. Sanity Check
2. Verify Configuration Settings
3. Test Packet Filtering between Interfaces
4. Test Reachability between VLANs

Each of these steps is described in more detail below.

5.1.1 Step 1 – Sanity Check

Before starting the audit, perform a couple sanity checks to make sure you know what you're doing.

- Physically inspect the network cabling and interface ports. Make sure the correct interfaces are connected to the correct devices, for example, that ethernet0 is connected to a CSU/DSU, that ethernet1 is connected to the other border router, etc. If the network wiring is wrong (or confusing), then the rest of the audit will be invalid and time wasted.

- Check to make sure the correct settings are loaded into the router. It would be a shame to spend days auditing the router only to discover they are not the final settings. To check the router's configuration:

    1. Copy the current running-config using "`copy run tftp`".
    2. Compare this running-config against the backup configuration made in §4.3.4 above using the Unix command "`diff`" or "`xdiff`".
    3. If the running-config appears to be different, the reload the saved-config using "copy save run" and compare again.
    4. If it is still different, then reload the configurations using the steps described in §4.3 above.

[17] SEI OCTAVE (Operationally Critical Threat, Asset, and Vulnerability Evaluation) is an excellent example on how to perform an enterprise security audit and risk assessment.

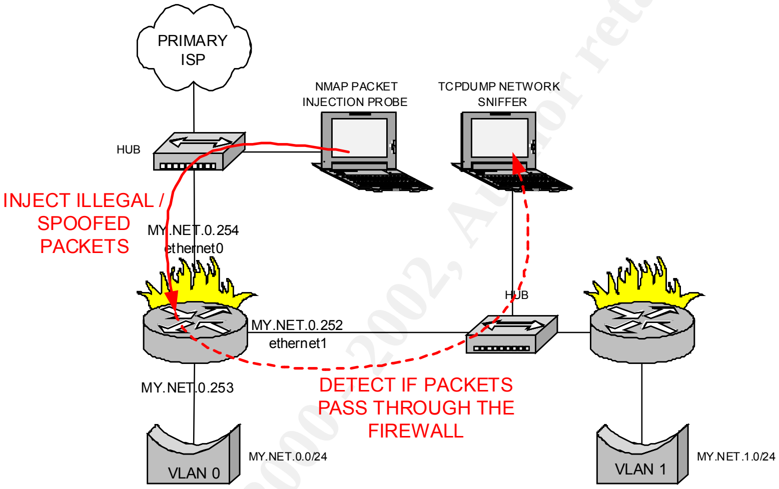
5.1.2 Step 2 – Verify Configuration Settings

Use the commands "`show access-lists`", "`show ip access-list`", "`show ip interfaces`", and "`show running-config`" to confirm the that each of the ACL rules is loaded and applied to the interfaces, as described in §4.5 above.

For additional verification, walk through the backup file of the running configuration to ensure that each of the ACL rules and lists are defined and applied in the correct order.

5.1.3 Step 3 – Test Packet Filtering Between Interfaces

For this step, configure two laptops[18]: one with Nmap to be used as an injection probe, and one with Tcpdump to be used as a network sniffer[19].



The injection probe is used to send illegal, spoofed packets into an interface of the router, and the network sniffer is used to monitor a different interface and detect if the packets were blocked by the firewall or allowed to pass through.

While a number of different, individual tests are performed on different combinations of interfaces, IP addresses, ports, and protocols, each test repeats the same basic steps:

1. Use the command "`clear access-list counters`" to clear the access list match counts (used in step 3 below).

2. Inject legal and/or illegal traffic into the interface.

[18] I'll generally reuse old corporate laptops for just this purpose. While users dump their old Pentium 133MHz laptops for the latest PIII 500MHz+ systems, I'll grab them up and configure them with Linux and network diagnostics and/or security software.

[19] Typically, I'll configure the network sniffer to execute using a stealth port (refer to Snort FAQ http://www.snort.org/docs/faq.html question "*How do I setup snort on a 'stealth' interface*") or an Ethernet interface that is not assigned an IP address. That way, the sniffer will detect all traffic on the net but is itself unreachable, undetectable, and guaranteed not to introduce any spurious packets onto the segment.

3. Use the command "`show access-lists`" (as described at the bottom of §4.5 above) to display which access control lists were executed and how many times. Check the match counts against the number of illegal packets sent to make sure they were all properly found and rejected.

4. Inspect the syslog to determine if packets are being properly dropped. Note: The syslog is not guaranteed to log an entry for every packet that is dropped. If too many syslog entries are being output at the same time, then some of the entries will be silently dropped. Otherwise, the router can become quickly overloaded with syslog messages, essentially opening the door to a DoS attack!

5. Finally, even if everything else reports that the packets are being filtered correctly, check the tcpdump sniffer as a sanity check to ensure they aren't sneaking through, somehow. (Use tcpdump filter rules as applicable to limit sniffing to the injected data.)

While it is impractical to test all permutations of attacks, source addresses, destination addresses and ports, and interfaces, the following tests represent a representative sample for validating configuration of the router's access control lists. For brevity, this chart only tests connectivity and filtering to the public server network (MY.NET.0-1.0/24) and the e-commerce network (MY.NET.10-11.0/24). It does not test the other networks, such as basic web hosting (MY.NET.20-21.0/24). Furthermore, since ACL rules are only applied to the external interface, the injector is always connected to ethernet 0.

Injector			**Sniffer Interface**	**Expected Firewall Action**
Action	**Source**	**Destination**		
Test ping (ICMP Request). (no access-list, therefore implicitly dropped)				
Ping sweep	100.0.0.1	MY.NET.0-11.0-255	ethernet 1	Drop all.
Test border router. (inet-router-in)				
SYN scan	100.0.0.1	MY.NET.0.254, ports 1-65535	N/A	Drop all.
SYN scan	THEIR.NET.1.254	MY.NET.0.254, ports 1-65535	N/A	Allow port 179. Drop all else.
UDP scan	THEIR.NET.1.254	MY.NET.0.254, ports 1-65535	ethernet 1	Drop all.
Test LSLB. (inet-public-in)				
SYN scan	100.0.0.1	MY.NET.1.1, ports 1-65535	ethernet 1	Allow ports 80 & 443. Drop all else.
UDP scan	100.0.0.1	MY.NET.1.1, ports 1-65535	ethernet 1	Drop all.
hping2 SYN flood	100.0.0.1	MY.NET.1.1, port 80	ethernet 1	Allow all.
Test DNS servers. (inet-public-in)				
SYN scan	100.0.0.1	MY.NET.1.249, ports 1-65535	ethernet 1	Drop all.
UDP scan	100.0.0.1	MY.NET.1.249, ports 1-65535	ethernet 1	Allow port 53. Drop all else.
Test illegal source IP addresses. (inet-illegal-addr-in)				
SYN scan	10.0.0.1	MY.NET.1.1, ports 1-65535	ethernet 1	Drop all.
SYN scan	10.255.255.255	MY.NET.1.1, ports 1-65535	ethernet 1	Drop all.
SYN scan	172.16.0.1	MY.NET.1.1, ports 1-65535	ethernet 1	Drop all.
SYN scan	172.31.255.255	MY.NET.1.1, ports 1-65535	ethernet 1	Drop all.
SYN scan	192.168.0.1	MY.NET.1.1, ports 1-65535	ethernet 1	Drop all.
SYN scan	192.168.255.255	MY.NET.1.1, ports 1-65535	ethernet 1	Drop all.
SYN scan	224.0.0.1	MY.NET.1.1, ports 1-65535	ethernet 1	Drop all.
SYN scan	255.255.255.255	MY.NET.1.1, ports 1-65535	ethernet 1	Drop all.
Test broadcast addresses. (no access-list, therefore implicitly dropped)				
SYN scan	100.0.0.1	MY.NET.1.0	ethernet 1	Drop all.
SYN scan	100.0.0.1	MY.NET.1.255	ethernet 1	Drop all.
SYN scan	100.0.0.1	MY.NET.11.0	ethernet 1	Drop all.

Injector			Sniffer Interface	Expected Firewall Action
Action	**Source**	**Destination**		
SYN scan	100.0.0.1	MY.NET.11.255	ethernet 1	Drop all.
Test SYN flood protection. (PIX embryonic limit feature.)				
hping2 SYN flood	100.0.0.1	MY.NET.1.0	VLAN 11	Allow first 100 SYNs. PIX handles all else.
Test web servers. (inet-ecommerce-in)				
SYN scan	100.0.0.1	MY.NET.11.1, ports 1-65535	VLAN 11	Allow ports 80 & 443. Drop all else.
UDP scan	100.0.0.1	MY.NET.11.1, ports 1-65535	VLAN 11	Allow ports 80 & 443. Drop all else.
Test VPN. (inet-ecommerce-in)				
IP protocol scan	100.0.0.1	MY.NET.11.200	VLAN 11	Allow UDP (6) and IKSEC (50) only.
SYN scan	10.0.0.1	MY.NET.11.200	VLAN 11	Drop all.
UDP scan	10.0.0.1	MY.NET.11.200	VLAN 11	Allow 500. Drop all else.
Test different scanning mechanisms.				
SYN scan	100.0.0.1	MY.NET.1.1, ports 1-65535	ethernet 1	Allow ports 80 & 443. Drop all else.
FIN scan	100.0.0.1	MY.NET.1.1, ports 1-65535	ethernet 1	Allow ports 80 & 443. Drop all else.
Xmas scan	100.0.0.1	MY.NET.1.1, ports 1-65535	ethernet 1	Unknown!
Null scan	100.0.0.1	MY.NET.1.1, ports 1-65535	ethernet 1	Unknown!
Ack scan	100.0.0.1	MY.NET.1.1, ports 1-65535	ethernet 1	Allow ports 80 & 443. Drop all else.
Frag scan	100.0.0.1	MY.NET.1.1, ports 1-65535	ethernet 1	Allow ports 80 & 443. Drop all else.
Frag scan	100.0.0.1	MY.NET.11.1, ports 1-65535	VLAN 11	Allow ports 80 & 443. Drop all else.
Test misc. packet oddities (misc. hardening configurations)				
hping2 mask request	100.0.0.1	MY.NET.0.254, ports 1-65535	N/A	Drop all.
hping2 source route IP packets	100.0.0.1	MY.NET.1.1	ethernet 1	Drop all.

While this may seem like a lot of tests (and it is!), by planning them out in advance and preparing, it is possible to run though all of these tests (in the chart above) in a single day.

Risks: A valid, publicly accessible IP address is needed to perform many of these tests. The table above arbitrarily selected the address “100.0.0.1”. However, be advised that any responses to the scanning (and there ***WILL*** be responses) will be sent over the Internet to the true owner of 100.0.0.1! Therefore, select this address wisely, and if you cannot locate an unassigned address, then please use one that you already own or borrow one from a friend (and let them know first, before “borrowing” it!).

5.1.4 Step 4 – Test Reachability between VLANs

Finally, test the connectivity between VLANs to ensure that traffic is being routed properly and not shunted between VLAN networks.

Start by passively sniffing each of the VLANs *while the network is in use*, looking for packets with illegal addresses. The following is a lists invalid destination IP addresses to be seen for each of the VLANs:

Network	Illegal Destination IP Addresses
VLAN 0	10.0.0.0/8
VLAN 1	10.0.0.0/8
VLAN 2	Anything other than 10.0.2.0/24
VLAN 3	Anything other than 10.0.3.0/24
VLAN 10	10.0.0.0/8 and MY.NET.0.0/16 *except for* MY.NET.10.0/24
VLAN 11	10.0.0.0/8 and MY.NET.0.0/16 *except for* MY.NET.11.0/24
VLAN 12	Anything other than 10.0.12.0/24
VLAN 13	Anything other than 10.0.13.0/24

Network	Illegal Destination IP Addresses
VLAN 14	Anything other than 10.0.14.0/24
VLAN 15	Anything other than 10.0.15.0/24
VLAN 16	Anything other than 10.0.16.0/24
VLAN 17	Anything other than 10.0.17.0/24

Then actively attempt to send packets between disconnected VLANs, testing for leaks. While there are a great number of permutations, it is possible to test them all quickly as follows:

1. Run a script that continuously sends hping2 packets to each of the VLANs. Spoof the source address to use 172.16.123.123, so the packets can be easily detected. For example:

```
#/bin/csh

while (1)
  # Send TCP ack port 80 to each VLAN
  hping2 -p 80 -A -a 172.16.123.123 MY.NET.0.211
  hping2 -p 80 -A -a 172.16.123.123 MY.NET.1.211
  hping2 -p 80 -A -a 172.16.123.123 10.0.2.211
  hping2 -p 80 -A -a 172.16.123.123 10.0.3.211
  hping2 -p 80 -A -a 172.16.123.123 MY.NET.10.211
  hping2 -p 80 -A -a 172.16.123.123 MY.NET.11.211
  hping2 -p 80 -A -a 172.16.123.123 10.0.12.211
  hping2 -p 80 -A -a 172.16.123.123 10.0.13.211
  hping2 -p 80 -A -a 172.16.123.123 10.0.14.211
  hping2 -p 80 -A -a 172.16.123.123 10.0.15.211
  hping2 -p 80 -A -a 172.16.123.123 10.0.16.211
  hping2 -p 80 -A -a 172.16.123.123 10.0.17.211

  !! Ping each VLAN
  hping -1 -a 172.16.123.123 MY.NET.0.211
  hping -1 -a 172.16.123.123 MY.NET.1.211
  hping -1 -a 172.16.123.123 10.0.2.211
  hping -1 -a 172.16.123.123 10.0.3.211
  hping -1 -a 172.16.123.123 MY.NET.10.211
  hping -1 -a 172.16.123.123 MY.NET.11.211
  hping -1 -a 172.16.123.123 10.0.12.211
  hping -1 -a 172.16.123.123 10.0.13.211
  hping -1 -a 172.16.123.123 10.0.14.211
  hping -1 -a 172.16.123.123 10.0.15.211
  hping -1 -a 172.16.123.123 10.0.16.211
  hping -1 -a 172.16.123.123 10.0.17.211

  # Etc.  Insert other hping2 commands that you think might
  # breach the VLANs.

  # Sleep for a second before the next sweep, to prevent
  # flooding the network.
  sleep 1
end
```

2. Run tcpdump to search for packets sent from address 172.16.123.123, as follows:

```
tcpdump ‘src host 172.16.123.123’
```

3. Plug the injector into the VLAN 0 and plug the sniffer into VLANs 1 though 17, in order. Wait 2-3 seconds on each VLAN before disconnecting the sniffer and moving it to the next VLAN. Then connect the injector onto VLAN 1 and repeat. Etc. The entire procedures takes approximately 30-60 minutes, depending upon how fast (and accurate!) you are at plugging and unplugging network connectors.

5.2 COST ESTIMATES FOR THE AUDIT

The following is an estimate of costs (time and equipment) for testing the left border router (i.e., the primary firewall), as described in §5.1 above, to ensure the security policies are configured and operating correctly.

Task	Calendar Days
Developing and documenting the test plan.	1 day
Peer review of network design and test plan.	2 days
Sanity check and verify configuration settings (steps 1 & 2)	1 day
Prepare to test packet filtering between interfaces (step 3)	1/2 day
Test packet filtering between interfaces (step 3)	3 days
Test reachability between VLANs (step 4)	1/2 day
Analyze and writeup test results and recommendations	2 days

Therefore, the total estimate is 2 man-weeks[20] to complete the audit, from conception to the final report. Of course, the estimate is significantly longer for the entire network, but then again, that's a pretty big job!

The audit requires no additional equipment to perform. Therefore, the only costs are in terms of staffing resources and not capital purchases or additional operating expenses.

5.3 TEST RESULTS

The tests revealed a minor bug in the access control rules. The list "inet-ecommerce-in" contains the following two rules:

```
!! Allow HTTP & HTTPS to web servers (MY.NET.10-11.1-127)
permit tcp any MY.NET.10.0 0.0.1.127 eq 80
permit tcp any MY.NET.10.0 0.0.1.127 eq 443
```

As stated in the comments, these rules are intended to allow HTTP & HTTPS traffic to addresses MY.NET.10-11.1-127. However, the *implementation* of these rules allows traffic to MY.NET.10-11.0 as well, which is interpreted as a broadcast port by some operating systems.

Therefore, in order to plug this hole, the following rule is inserted before the two permits:

```
deny tcp any MY.NET.10.0 0.0.1.0
```

Other than that one minor glitch, the implementation passed the remaining tests from the audit and was approved to go live.

[20] I really don't know what the politically correct term is to replace "man-weeks". "Person-weeks" or "staff-weeks", perhaps. Personally, I'm fond of the term "geek-weeks". ☺

6 DESIGN UNDER FIRE

I have selected the network design by Kevin Olree on May 10, 2001 for vulnerability analysis and ethical intrusion. His paper is available at http://www.sans.org/y2k/practical/Kevin_Olree_GCFW.doc.

His network uses a PIX firewall to create two networks:

- A "high-security" internal network containing the fortune database, as well as user workstations, mail servers, and other back-office systems.
- A "medium-security" service network, containing a publicly-accessible web server and mail proxy.

This PIX works double-duty, acting as both a firewall and a VPN gateway.

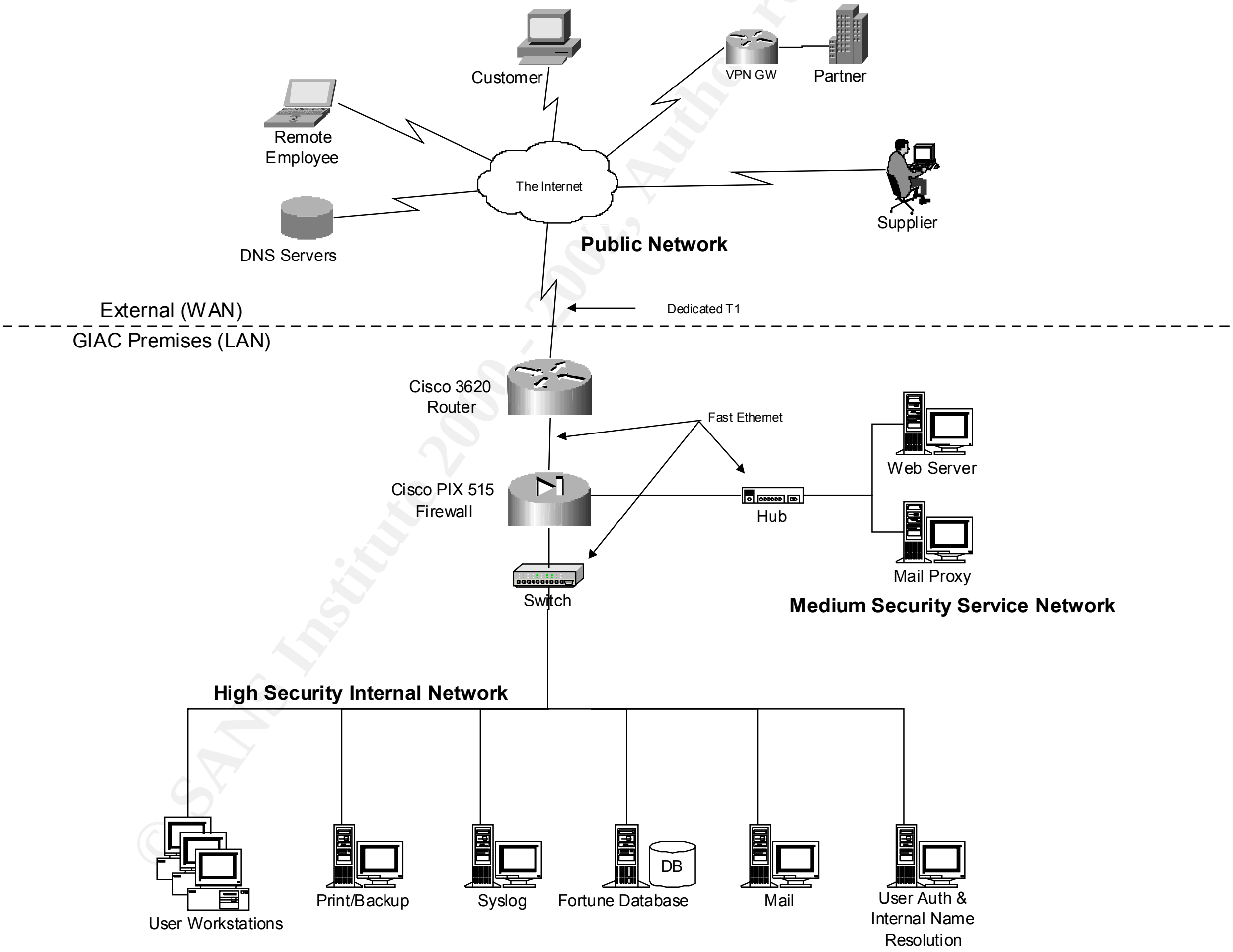


Figure 1.2: Physical Network Design

6.1 ATTACK THE PIX

6.1.1 Choosing a Vulnerability

The CVE dictionary [7] reports seven known vulnerabilities for the PIX firewall:

Name	Description
CVE-1999-0157	Cisco PIX firewall and CBAC IP fragmentation attack results in a denial of service.
CVE-1999-0158	Cisco PIX firewall manager (PFM) on Windows NT allows attackers to connect to port 8080 on the PFM server and retrieve any file whose name and location is known.
CVE-2000-0613	Cisco Secure PIX Firewall does not properly identify forged TCP Reset (RST) packets, which allows remote attackers to force the firewall to close legitimate connections.
CVE-2000-1022	The mailguard feature in Cisco Secure PIX Firewall 5.2(2) and earlier does not properly restrict access to SMTP commands, which allows remote attackers to execute restricted commands by sending a DATA command before sending the restricted commands.
CVE-2000-1027	Cisco Secure PIX Firewall 5.2(2) allows remote attackers to determine the real IP address of a target FTP server by flooding the server with PASV requests, which includes the real IP address in the response when passive mode is established.
CAN-1999-1100	** CANDIDATE (under review) ** Cisco PIX Private Link 4.1.6 and earlier does not properly process certain commands in the configuration file, which reduces the effective key length of the DES key to 48 bits instead of 56 bits, which makes it easier for an attacker to find the proper key via a brute force attack.
CAN-2001-0375	** CANDIDATE (under review) ** Cisco PIX Firewall 515 and 520 with 5.1.4 OS running aaa authentication to a TACACS+ server allows a remote attacker to cause a denial of service via a large number (approximately 426) of authentication requests.

According to Cisco's product security advisories [5], each of these vulnerabilities have resolved by release 6.1(1) of the PIX software. Analyzing each of these vulnerabilities against the target network's design reveals:

- CVE-1999-0157 is an old vulnerability that was corrected in 1998 by PIX software release 4.2(2). The target network was setup in early 2001 and therefore is not vulnerable to it.

- CVE-1999-0158 is an old vulnerability that was corrected in 1998 by PIX software release 4.2(2). The target network was setup in early 2001 and therefore is not vulnerable to it.

- The target network **may be vulnerable** to CVE-2000-0613 if the PIX software has not been updated to version 4.4(5) or 5.1(2) or greater, both of which were released on June 9, 2000.

- The target network is not vulnerable to CVE-2000-1022 since it does not use utilize the SMTP command filtering feature ("fixup protocol smtp 25").

- The target network is not vulnerable to CVE-2000-0127 since it does not use "`fixup protocol ftp`".

- CAN-1999-1100 is an old vulnerability that was corrected in 1998 by PIX software release 4.2.1. The target network was setup in early 2001 and therefore is not vulnerable to it.

- The target network is not vulnerable to CAN-2001-0375 since it does not use AAA authentication or a TACACS+ server.

Thus, the only known vulnerability that the target network may be susceptible to is CVE-2000-0613, or forged TCP reset packets used to close legitimate connections.

6.1.2 Designing the Attack

According to Cisco's advisory on CVE-2000-0613:

> *"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. Data exchange based on any other protocol is not affected."* [14]

Therefore, all we need to know are four values in order to exploit this vulnerability: the source IP address, source port, destination IP address, and destination port for a TCP connection. Given reconnaissance and social engineering, it should be possible to determine the IP addresses for the public web server (9.9.9.4), VPN gateway (9.9.9.2), and remote VPN routers for the suppliers (4.4.4.1, 4.4.4.2, and 4.4.4.3). IKE communications is sent to a well-known port. That only leaves one value left to guess: the (ephemeral) source port. Since there are only 65535 possible values, at worst this can be brute forced.

The command "hping –R" is used to spoof TCP RST packets.

6.1.3 Results from the Attack

Before proceeding, a test is performed to determine if the PIX is vulnerable to the attack or if it has been properly patched. The test proceeds as follows:

1. A TCP connection is made to the web server, but the connection is held open after the 3-way handshake completes.

2. Since we made the connection, we know all four values (src & dst addr & port) necessary to perform the attack. The command:

   ```
   hping2 –R –s <src-port> -a <src-ip> –p <dst-port> <dst-ip>
   ```

   sends a properly spoofed TCP RST packet to the web server.

Unfortunately, the attack is not successful, indicating that the PIX is running software release 5.1(2) or later.

6.2 DDoS Attack

There are a number of different exploits for performing DDoS (distributed denial of service) attacks. Two common tools described in SANS Track 4 (Incident Handling & Advanced Hacker Exploits) are TFN2K (Tribal Flood Network 2000) and Trin00.

For this example, I assume that 50 cable/DSL systems have been compromised and configured as TFN2K servers. (TFN2K is available at http://www.angelfire.com/rock/nsi). Triggered by an ICMP echo reply packet, the 50 TFN2K servers flood the target web server at 9.9.9.4with SYN packets.

According to the Kevin's practical §2.4.2, access to the web server has been configured on the PIX as follows:

```
static (service,outside)9.9.9.4 10.2.2.4 netmask 255.255.255.255 1000
```

The value "1000" at the end of the "static" sets the maximum number of connections allowed through the static to 1000. Once the limit is reached, the PIX box will drop all new connection attempts.

Once launched, the TFN2K attack quickly exceeds 1000 connections, essentially causing the PIX to shut down all further communications with the web server *even though* the web server itself might not be vulnerable to SYN floods (such as via SYN cookies)!

The static command has an additional option called the "embryonic connection limit" that can be specified after the "maximum number of connections" parameter. For example, the command:

```
static (service,outside)9.9.9.4 10.2.2.4 netmask 255.255.255.255 1000 100
```

set the embryonic limit to 100. This feature is discussed in more detail in §3.2.2 above. In this example, setting max_conns to 1000 and em_limit to 100 results in the following behavior for a SYN flood:

- The first 100 SYN packets are allowed through to the web server, resulting in 100 half-opened TCP connections.
- All additional SYN packets are intercepted by the PIX box. Rather than dropping the SYN packets altogether, the PIX sends a SYN/ACK and awaits a reply. Any connections that complete the 3-way handshake are then (and *only* then) sent to the web server.

Therefore, using embryonic connection limits, the SYN packets from the attack are ultimately culled and valid SYN connections are allowed to connect, essentially minimizing the effect of the TFN2K attack.

Ironically, Kevin intended to use the embryonic connection limit all along, stating:

> *"The last parameter in the static commands specifies an 'embryonic' connection limit of 1000 for the web server and 50 for the mail proxy server. This tells the PIX to disallow new connection requests if the number of incomplete connections has reacheed this limit. Ideally we have set these values high enough to accommodate the peak traffic load while still protecting the hosts from SYN attacks."*

However, his "static" command was misconfigured and accidentally specified the "maximum number of connections" value instead of the "embryonic connections limit", thus opening the servers up to DDoS attack.

6.3 COMPROMISING AN INTERNAL SYSTEM

There are a number of interesting attack scenarios for compromising an internal system.

The network is designed with an "all or nothing" approach… there is little additional protection past the primary firewall. If the firewall can be compromised or bypassed, then a hacker would have access to *everything*!

All it takes is finding *one* method of bypassing the firewall, and there appear to be many to choose from.

6.3.1 Scenario 1 – Attack the Web Server

The CVE dictionary [7] reports 84 known vulnerabilities for IIS, as indicated using http://www.cve.mitre.org/cgi-bin/cvekey.cgi?keyword=iis[22]. Scan the server with vulnerability scanner such as nessus or whisker.

Once the web server is compromised, then a hacker may:

- (Easy) Modify the web site to collect user passwords.
- (Easy) Access the database and download all its data, including: the complete library of fortunes for sale, the customer database, usernames and passwords (for online customers), etc.
- (Harder) Access port tcp/1521 of the database server (Oracle) and attempt to compromise that system next.

Note: It is likely that employees of GIAC Enterprises have login accounts to the customer web site. Furthermore, many employees will use the same username and password for their internal user accounts, the commerce web site, and their remote VPN login accounts.

[21] By the time you follow this link, the list of vulnerabilities will undoubtedly be even longer!

[22] By the time you follow this link, the list of vulnerabilities will undoubtedly be even longer!

Therefore, since the VPN login accounts for employees are not configured to require connection from a specific source IP address, attacking the web server and downloading the username/password list may be an expedient method of bypassing the firewall and cracking the entire network!

Or, if the Oracle database server isn't properly hardened, it may be possible to execute commands remotely on it. (Microsoft SQL Server has similar remote execution "features".)

6.3.2 Scenario 2 – Attack the Users

Employee workstations are connected to the "high security" network. If one of these workstations (or laptops) is compromised, then it can be used to compromise the entire network!

- One possibility is to exploit a vulnerability in Internet Explorer. The CVE dictionary [7] reports 101 known vulnerabilities for IE, as indicated using http://www.cve.mitre.org/cgi-bin/cvekey.cgi?keyword=ie. For example, an ActiveX attack can result in complete control over the computer.

- Or, get an employee to access a web site and accept an unsigned ActiveX control that is actually a Trojan.

- Or, get the employee to load Trojan software. For example,

    - Wrap a backdoor tool (such as Loki or Inverse WWW Telnet) around something cool or interesting (such as a cracked game or free software tools) using Silk Rope 2000.

    - Or e-mail them an executable attachment from a spoofed letter that appears to be written by their boss, or (better yet) by someone in the IS department instructing them to execute the attachment. (The IS group at my old company did that all the time!)

    - Or e-mail them an executable attachment from a headhunter or reputable recruiting source, such as Monster.com or Hot Jobs.

    - Or pretend you're a salesman and send the Director of I.S. an e-mail containing a product demo of your new "anti-spam" software. Then spam the hell out of their site and see if they take the bait.

- Or, wardial the company's exchange and hope you find a user's workstation setup with an auto-answer modem and PC Anywhere.

The possibilities for social engineering are endless!

6.3.3 Scenario 3 – Attack the Partner's Network

If a hacker can compromise one of the partner's networks, then their dedicated LAN-to-LAN VPN gateway can be used to crunch through the firewall's hard shell and enjoy the soft center of the "high security" network.

6.3.4 Chosen Attack

I would choose scenario 2 (attack the users) as the path of least resistance.

1. First, configure a BO2K server configured as an unsigned ActiveX control.

2. Design a fake website that appears like a new headhunter site for high-tech jobs, and incorporate the BO2K ActiveX control into the site.

3. Using a compromised system on the network (such as one of the 50 systems compromised with TFN2K in §6.2 above), setup the fake web site and make it publicly accessible from the Internet.

4. Send e-mail to employees at GIAC Enterprises advertising the new job-recruiting site. Make it enticing. For example, perhaps claim to be an internal recruiting department within a nearby aerospace company that is going through a growth spurt and eagerly looking to hire new employees.

5. Then sit back and wait for someone to take the bait. As soon as anyone in the company accepts the ActiveX control, then BO2K has control of their system.

6. Hopefully, the ActiveX control was accepted by a system administrator, since they tend to have administrative access from their computers, making it easier to take over the rest of the computers on the "secure network". Otherwise, it may take a little work, but ultimately the entire network will be broken within the next day or two.

7 BIBLIOGRAPHY

7.1 Reference Materials

1. *Cisco IOS Configuration Guides and Command References, Release 12.1*, http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/index.htm , Cisco.

2. *Cisco IOS Firewall Overview*, http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/secur_c/scprt3/scdfirwl.htm#xtocid224565, Cisco.

3. *Cisco IOS Security Configuration Guide, Release 12.1*, http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/secur_c/index.htm, Cisco.

4. *Cisco PIX Firewall Online Documentation*, http://www.cisco.com/univercd/cc/td/doc/product/iaabu/pix/index.htm, Cisco.

5. *Cisco PSIRT (Product Security Incident Response Team) Advisories*, http://www.cisco.com/warp/public/707/advisory.html, Cisco.

6. *Cisco VPN 3000 Concentrator, Release 3.1*, http://www.cisco.com/univercd/cc/td/doc/product/vpn/vpn3000/3_1/index.htm, Cisco.

7. *Common Vulnerabilities and Exposures*, http://www.cve.mitre.org, The MITRE Corporation.

8. *SANS Track 2 module 2.3.1 - Defense in Depth – Routers*, SANS Institute.

7.2 Quotes

9. *Cisco IOS IP and IP Routing Command Reference, Release 12.1 – "access-list (IP extended)" command*, http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/ip_r/iprprt1/1rdip.htm#xtocid132652, Cisco.

10. *Cisco IOS Security Configuration Guide, Release 12.1 – "Authentication, Authorization, and Accounting" chapter*, http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/secur_c/scprt1/index.htm, Cisco.

11. *Cisco IOS Security Configuration Guide, Release 12.1 – "Creating Access Lists" section,* http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/secur_c/scprt3/scdacls.htm#xtocid181688, Cisco.

12. *Cisco PIX Firewall Command Reference – "icmp" command,* http://www.cisco.com/univercd/cc/td/doc/product/iaabu/pix/pix_61/cmd_ref/gl.htm#xtocid121945, Cisco.

13. *Cisco PIX Firewall Command Reference – "static" command,* http://cio.cisco.com/univercd/cc/td/doc/product/iaabu/pix/pix_61/cmd_ref/s.htm#xtocid187320, Cisco.

14. *Cisco Secure PIX Firewall TCP Reset Vulnerability*, http://www.cisco.com/warp/public/707/pixtcpreset-pub.shtml, Cisco.

15. *Cisco VPN 3000 Series Concentrator Reference, Volume I: Configuration – "Authentication Servers" configuration,* http://www.cisco.com/univercd/cc/td/doc/product/vpn/vpn3000/3_1/config/usermgt.htm#xtocid758139, Cisco.

16. *Cisco VPN 3000 Series Concentrator Reference, Volume I: Configuration – "Base Group" configuration,* http://www.cisco.com/univercd/cc/td/doc/product/vpn/vpn3000/3_1/config/usermgt.htm#xtocid7584, Cisco.

17. *Cisco VPN 3000 Series Concentrator Reference, Volume I: Configuration – "IPSec Parameters' configuration,* http://www.cisco.com/univercd/cc/td/doc/product/vpn/vpn3000/3_1/config/usermgt.htm#xtocid75819, Cisco.

18. *Cisco VPN 3000 Series Concentrator Reference, Volume I: Configuration – "IPSec SAs" section,* http://www.cisco.com/univercd/cc/td/doc/product/vpn/vpn3000/3_1/config/polmgt.htm#xtocid1214644, Cisco.