Global Information Assurance Certification Paper

Perimeter Network Design for GIAC Enterprises, Inc.

GIAC Certified Firewall Analyst (GCFW)

Practical Assignment

Version 2.0 (Revised May 26, 2003)

By Daniel J. MacDonald

Submitted December 29, 2003

Table of Contents

Abstract

GIAC Enterprises (GIACE) is an international provider of fortune cookies that primarily conducts business via E-Commerce over the Internet. Consequently, GIACE requires strong security while concurrently allowing appropriate access across the Internet. This network design meets those needs. Four topics are contained as part of the design. First, the architecture section details the design process: enumerating the required business system and user access classifications, defining the design premises, describing the actual architecture chosen, and choosing the products for final implementation. Second, an implementation tutorial is provided, including explanations of options used for configuration of the perimeter firewalls, host firewalls, border routers, and internal routers. Third, verification of the perimeter firewall is performed, including planning the verification, the results of the verification, and discussion of recommendations stemming from the verification results. Finally, an alternate design for the GIACE perimeter is presented with three theoretical attacks: an attack against the firewall, a DDoS attack from the Internet, and an attack against a compromised system. Appendices are provided with full sample configurations for selected devices.

Background

GIAC Enterprises (GIACE) is an upstart dealer of fortune cookie sayings in the rapidly exploding international proverbial phrase market. Due to rapid growth in this segment, GIACE management wishes to ensure that the infrastructure for their new upstart is properly sized to ensure availability of all business systems. Since some of GIACE's key management figures were formerly employed by a certain U.S. bank during the SQL-Slammer incident in January 2003, management also recognizes the importance of availability and confidentiality of the data contained within GIACE systems as well. In short, management has provided leeway to build a secure infrastructure for GIACE from the ground up. The following objectives have been communicated from management (in no particular order):

- The infrastructure must be scalable. No specific benchmarks or targets are available to work from, however it is expected that extremely rapid growth will follow the opening of GIACE.
- The design must be sufficiently redundant to ensure high availability, even under periods of high load. Management has not authorized expenditure for geographically diverse data centers at this time, however may do so in the future.
- The infrastructure must contain sufficient controls to protect against and recover from data corruption, either accidental or malicious.
- Since GIACE deals exclusively in intellectual property, the design must protect GIACE's intellectual property from unauthorized disclosure.
- Like all online retailers, GIACE must maintain certain customer and billing records that will contain private 3rd party information. GIACE's Privacy Policy prohibits the disclosure of this information. Therefore, the design must protect against the unauthorized disclosure of private 3rd party information.
- While GIACE is not located in a state that has implemented specific legislation regarding data security, several other states are evaluating proposed legislation at this time. GIACE wishes to maintain a state of security that will ease compliance if such legislation is passed within GIACE's jurisdiction.

As this is a new organization, exact personnel requirements are unknown at this time; however, GIACE projecting at this point that this will be a medium-sized organization of approximately 150 employees at a single location. GIACE intends to hire support and operational staff with awareness and average competency in security topics.

Definitions

Prior to embarking on this design, terminology should be defined to ensure consistency and clarity. Industry standard terms sometimes carry different meanings in different contexts. Therefore, for the purposes of this document, the following terms are defined:

- **System**: a server, workstation, device or any grouping thereof that fulfills a particular business function.
- **Cluster**: a group of two or more identically functioning servers.
- **Server**: one particular machine providing services to users. In most cases, a server will be a component of a larger system. A server shall generally not perform workstation functions as defined below.
- **Workstation**: a machine that does not provide services to users, but instead allows a user to access services offered on the network. A workstation shall generally not perform server functions as defined above.

Throughout this document, servers and devices that are mission critical shall have redundancy and therefore are referred to in the plural or as a "cluster". Servers and devices that are not mission critical, or for which clustering is not viable, are referred to in the singular. Non-redundant systems may be abstracted to more than one system where required, however the systems will not be identically functioning in nature. For example, multiple file servers may be deployed, however they will not all contain identical data and therefore are not interchangeable.

Security Architecture

No implementation can be successful without proper planning and design. Business sometimes requires organic growth and expansion, but even in those cases, expansion and implementation performed in a vacuum will give less than satisfactory results. To begin the project for GIACE, we must first determine the required business systems, goals of the implementation and user classifications, then we can determine the traffic flows dictated by the users to access the required systems. Only after that can we identify the necessary controls and architecture to provide an appropriate solution to fit the traffic flows and the goals of the organization. Finally, vendors may be identified to fit into the architecture puzzle.

Required Business Systems

GIACE is a fully functioning e-business; therefore it has all the requirements of a typical e-business. Systems utilized by GIACE are classified as follows:

- **File/Print**: A server that provides typical file and print services. These servers will also provide time and name resolution services for GIACE.
- **Groupware**: A server that provides scheduling, email and other groupware functions for internal users. This server also provides time and name resolution services for GIACE.
- **Informational website**: This system hosts the publicly available informational website. It primarily contains static content and does not interface with any outside systems other than to send email notifications of system status and website visitors.
- **E-Commerce website**: This system forms the interface through which customers buy fortunes in bulk. An "exchange" system used by suppliers and partners for transfer of bulk fortunes is also hosted on this system due to the similar nature of the applications. This system is comprised of a cluster of front-end web servers to host the web applications and a certificate server for assignment of certificates to vendors and partners. This system also interfaces with the fortunes system below.
- **Client management**: This system hosts the Customer Relationship Management (CRM), Bookkeeping, Accounts Receivable, and Accounts Payable applications. Since these applications are for internal use only, fat client interfaces have been developed. Therefore, the only servers required are for the database cluster.
- **Fortunes**: This database cluster hosts the database of fortunes. Since the product sold by GIACE is data, separate inventory tracking is not required. This system has a fat-client interface for internal use only and also interfaces with the E-Commerce website.
- **Internet Access**: Servers that facilitate Internet access not covered by a system defined above. This system is comprised of a cluster of proxy servers that also serve as an email gateway. These services also provide time and name resolution services for the service DMZ hosts as defined below. Public DNS for GIACE

domains is not hosted onsite, but is delegated across ISPs providing Internet connectivity for GIACE.

- **Remote Access**: Servers or devices that facilitate access by trusted users from outside the trusted network. Primarily consists VPN access and Terminal Services.
- **Administrative systems**: Servers or workstations that facilitate monitoring and management of the GIACE network environment. Functions include IDS, logging, backups and authentication.

These are the only systems identified at this time as necessary for business purposes at GIACE. No systems or services are allowed that have not been identified by GIACE Senior Management as necessary for business purposes. Therefore, any systems beyond those defined above must have the approval of GIACE Senior Management.

User Classifications

The users at GIACE fall into one of the following six classifications (from least controlled access to most controlled access):

- **Internal Users**: GIACE employees or contractors physically connected directly to the GIACE network. Internal users are allowed full access to all internal systems for which they are assigned permissions. Internal users are **not** allowed to interface directly with the Internet, but must use an intermediate proxy. Certain groups of Internal users (particularly those with administrative rights) may be required to negotiate IPSec encryption for access to certain applications or servers.
- **Mobile Users**: GIACE employees located outside of the physical GIACE network, but requiring greater use of internal systems. Mobile users will be required to connect via an encrypted VPN to access internal systems. Due to bandwidth limitations for mobile users, a terminal server is available only after authentication to the VPN.
- **Partners**: Representatives of companies that provide translation services or resell GIACE fortunes. Partners may only access the fortune database via a web application hosted on the E-Commerce system. Partners may upload completed fortune sets or download fortunes that have been released to them by an authorized GIACE employee. All uploaded fortunes are held in a separate area for check in by authorized GIACE employees. The E-Commerce system will communicate with the Customer Management database via logic within the fortune database to properly record accounting information.
- **Suppliers**: Representatives of companies that supply GIACE with fortune sets. Suppliers may only access the Fortune database via a web application hosted on the E-Commerce system. Suppliers may upload fortune sets via the same web application to a separate area for check-in by authorized GIACE employees. The E-Commerce system will communicate with the customer management database via logic within the fortune database to properly record accounting information.

- **Customers**: Purchasers of fortune sets. Customers may only access the fortune database via a web application hosted on the E-Commerce system. The E-Commerce system will communicate with the customer management database via logic within the fortune database to properly record billing information.
- **General Public**: The general public will be allowed to browse the Informational Website only. The public may become customers by entering the E-Commerce site.

In some cases, a user may fall into more than one classification, in which case care must be taken to ensure that the user fulfills the requirements of only the appropriate classification. This may be done automatically by determining the user's physical location or manually by providing additional or alternate identification and/or authentication. For example, a "Supplier" is a member of the "Public Users" category until they authenticate to the E-Commerce site using their "Supplier" level credentials, at which point the user may exercise the abilities afforded the "Supplier" classification.

Note that these are general classifications. More granular control may (and should) be exercised over each user's privileges within the GIACE network.

Defined Traffic Flows

This section will define the acceptable traffic flows within the GIACE network. This section will enumerate all traffic flows required for the business operation of GIACE; therefore, all other traffic shall be considered suspect. GIACE generally follows the best-practice of deny unless specifically allowed; therefore, all traffic not contained in a class defined here shall be denied. Traffic flows are defined on a system-by-system basis. Understanding these traffic flows is paramount to understanding the security device policy configurations to follow.

The following traffic flows are common to all systems:

- All GIACE systems shall be configured to send logging information to the logging server. IPSec AH signing shall be used whenever supported.
- All internal GIACE systems shall be configured to receive time and name resolution services from the file/print servers and the groupware server. DMZ servers shall receive time and name resolution from the proxy servers. Designated time and name resolution servers shall be the only servers allowed to receive time or name resolution information from the Internet, and then only from specifically designated upstream servers.
- All GIACE systems shall be allowed to connect to the database services on the database clusters, except for servers in the DMZ. The E-Commerce web servers in the DMZ will be allowed to connect only to the fortunes database cluster to provide E-Commerce services.
- All GIACE systems shall be allowed to connect to the proxy services on the proxy cluster for Internet access. This includes outbound connections from DMZ servers

for patch retrieval and other administrative functions. The proxy server will be used to limit Internet access on a user and/or source machine basis to the minimum access necessary at GIACE management's discretion. This is to limit damage in case of a security incident on the GIACE network.

- All GIACE systems shall be able to connect to DMZ web services (via the proxy servers) to which connections are allowed from the Internet, namely the informational and E-Commerce websites, and the CA website.

System traffic flows are defined as follows:

- **File/Print system:** This system provides basic file/print services to internal users and mobile users via the VPN. Clients and the terminal servers are allowed to connect to all file/print sharing ports, time services, AD authentication, and name resolution. One of the disadvantages to Windows products is that this is a very large exposure and difficult to control. Consequently, these servers will have HIDS installed.
- **Groupware:** This system provides e-mail and scheduling services to GIACE users. Clients are allowed to talk directly to mail services on this server, as well as time, AD authentication, and name resolution services. This server has some of the same drawbacks as the file/print servers mentioned above, therefore the same HIDS precaution will be taken. This server is also responsible for sending and receiving Internet e-mail via the e-mail gateway provided by the VirusWall software on the proxy cluster. Therefore, SMTP must be allowed in both directions between the exchange and VirusWall servers.
- **Internal Workstations/Terminal Server:** These systems are workstations used by internal users. These systems will have connectivity to both of the database clusters, the file/print servers, and the proxy servers. Users will generally connect to the Terminal Server via the VPN, however they are not restricted from connecting from the internal network to provide flexibility on the network. Certain users may be provided thin clients instead of full workstations at the discretion of GIACE support staff.
- **Informational Website:** The only purpose of these servers is to serve static content to the public at large. Therefore, the only traffic allowed permanently is HTTP traffic from any location. As mentioned above, SSH may also be activated from the console on a temporary basis to assist with management and file transfer during maintenance periods; however SSH shall be disabled when not in use and always restricted to internal clients only.
- **E-Commerce Website:** The purpose of this cluster is to provide an exchange location for fortunes between GIACE and its partners, suppliers, and customers. Servers in this cluster will require database communication with the fortunes database cluster to transfer fortunes and customer management information (such as billing and payables) to and from the fortunes database. Obviously, since this is a web-based E-Commerce system, HTTPS must be allowed to these servers from

the internet at large. Again, SSH may be activated temporarily for maintenance periods and disabled when not in use for these servers.

- **Certificate Server:** This server will be integrated into the E-Commerce website in order to provide PKI certificates to authorized E-Commerce users. MS-CA provides a web interface for certificate services, therefore the only necessary communication flow for this server is HTTPS.
- **Client Management:** This database cluster contains the proprietary customer, partner and supplier information. Internal clients shall be able to access this database directly, as well as the fortune database system. This cluster shall not be accessed directly from the E-Commerce website; instead all relevant data from the E-Commerce site shall pass through the fortunes database cluster first. This will help to isolate the proprietary client data further from the internet in case of compromise of the E-Commerce systems. It will also provide a queuing point for E-Commerce transactions in case of downtime for the client management cluster. While all internal systems and VPN clients will have connectivity to the client management database system, the database system will contain additional access controls to limit individual users to only authorized access.
- **Fortunes:** The fortune database cluster contains only fortunes and transactional data in transit. Separate data space is provided within this system for fortunes recently checked in from GIACE partners, but not yet approved by GIACE staff. Communication flows consist of database access from internal clients, the client management database cluster, and the E-Commerce site. While all internal systems and VPN clients will have connectivity to the fortune database system, the system will contain additional access controls to limit individual users to only authorized access.
- **Network Devices:** Network devices (such as the switches and routers) shall have all in-band management disabled and all management shall occur from out-of-band serial connections.
- **Administrative Systems:** Systems primarily for management of the GIACE network shall be located on the administrative VLAN and will require IPSec for all administrative communication. All platforms support IPSec except for some of the network devices, so exceptions may be made for devices only when IPSec is not available. Specific systems policies are as follows:
    - **Monitor Console:** This server will monitor for performance, uptime, and other operational statistics. Due to the ease of forging UDP packets, IPSec shall authenticate all SNMP packets in Authentication Header (AH) mode. (The privacy afforded by Encapsulating Security Payload (ESP) is not required, and the performance penalty from ESP makes AH preferable.) All SNMP enabled devices shall be configured read-only and will be configured to not respond to packets which are not AH authenticated. For extended visibility to this server, users will be allowed to connect remotely, however IPSec in ESP mode will protect all remote connections to the workstation. This server will also contain a modem configured in dial-only mode to facilitate transmission of notifications should all Internet lines become

unavailable. This modem will not be configured to connect using a transport-mode protocol such as PPP or SLIP, but instead will accomplish notification by a terminal session only.

- **IDS Console:** This server will control the IDS sensors located throughout the network. Due to the security-centric nature of all traffic to and from this server, all traffic shall be protected by IPSec ESP mode. As with the monitor console, authorized users will be allowed to connect remotely to this station, however ESP shall be required for all remote connections.
- **Backup Server:** This server is responsible for backing up all systems considered critical by GIACE management. Most backups shall occur directly over the Fiber Channel SAN for performance reasons, however some systems (such as the database systems and the exchange server) do not lend themselves to file-level backup such as that afforded by this method. For record-level backups, all connections shall be initiated by the backup server and protected by IPSec in ESP mode in order to protect the privacy of data during the backup process. Best practices for tape rotation and storage shall be followed, however are beyond the scope of this document. Remote control for this server does not provide significant benefit, therefore is not allowed.
- **Token Authentication Server:** This server provides authentication and management services for the tokens used for remote user authentication. The Authenex solution uses RADIUS to authenticate certificates stored on the hardware device. RADIUS is a challenge-response protocol and is reasonably protected against interception, especially considering the use of certificates; however the importance of this information makes the small performance tradeoff for ESP acceptable. Due to the importance of this server to the security posture of GIACE, management shall occur from only the local console.
- **Logging Server:** This server serves to collect logging information from across the network. This system shall be selected by GIACE closer to implementation, but shall run on a Linux or Windows platform and must accept logging from Syslog.

Security Controls

A variety of categories of controls are available to assist in fulfilling the objectives listed above. Administrative and physical controls are beyond the scope of this document, however sound hiring practices, physically securing the network infrastructure, separation of duties, change control, and security awareness training will greatly increase the security posture of GIACE. A variety of technical controls will be discussed here to establish the best security practice of "defense-in-depth." In short, this concept does not allow the security of a system to rest entirely upon any single control, but uses a variety of controls in a layered fashion to ensure the availability, confidentiality, and

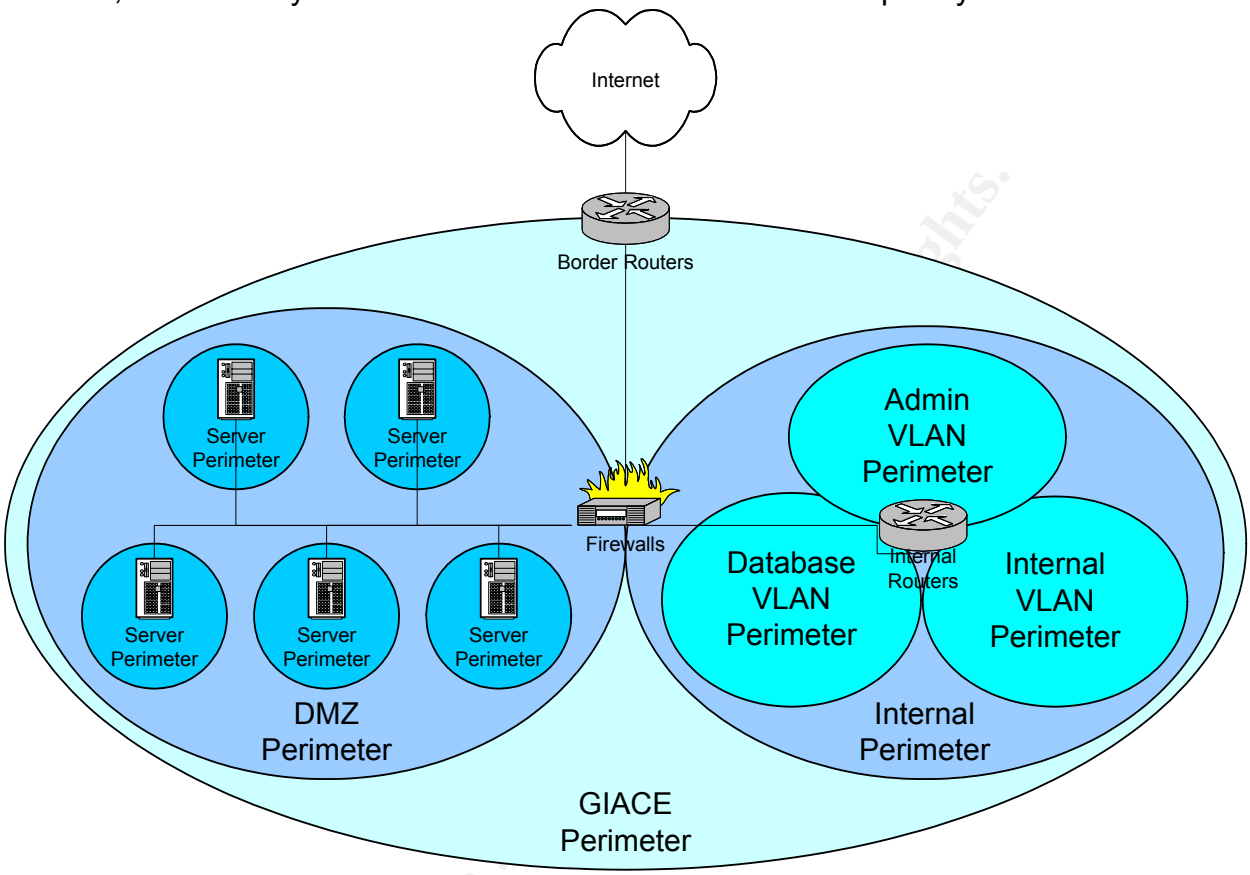
integrity of data contained in the GIACE network. The following technical controls are utilized within the design:

- **Redundancy**: Redundancy is built into mission-critical systems throughout the design. Data-lines, routers, firewalls, switches, SAN devices and critical servers are all configured with either active-active load balancing or active-standby failover to ensure that no one component can by itself cause an extended outage.
- **Traffic Control**: Traffic is limited to defined paths between systems that are reachable from the network. All critical internal systems allow only limited access to traffic flows necessary for required work processes. This traffic control takes place at subnet boundaries (whether defined physically or logically via VLAN) or system-by-system in the case of the Service DMZ. Traffic Control policy detail is defined in Security Policy and Tutorial below.
- **Monitoring/Intrusion Detection**: Sensors and monitoring stations are built into the network at key locations to detect operational and security events within the network. This provides visibility into anomalous occurrences within the GIACE network allowing for rapid and appropriate response to address threats to the availability, confidentiality, and/or integrity of data in the GIACE network as proactively as possible.
- **Strong Authentication**: Strong authentication is implemented at strategic points within the GIACE network. All perimeter access for Mobile Users requires the use of full two-factor authentication to eliminate the possibility of password attacks from outside the network perimeter. Due to the expense of implementation and management of a fully token-based system on the perimeter, GIACE will use certificate-based authentication for partners, suppliers, and wholesale customers. Individual customers will not be authenticated, using a one-time credit card collection for purchase. Online delivery will occur within the same session, negating the need to authenticate the user beyond the credit card.
- **Encryption**: Encryption is implemented on all non-public connections from outside the GIACE network to ensure privacy of all data transported to point outside the GIACE environment. (Members of the general public purchasing fortunes fall into the customer category throughout the transaction.) Due to the countermeasures implemented to prevent sniffing/interception attacks within the GIACE network, encryption may be limited to the perimeter of the network to allow for better monitoring visibility.
- **Cryptographic Signing**: Traffic signing is implemented on critical systems to ensure integrity of data. Signing shall be used only when privacy of the affected traffic is not critical, and the performance impact of full encryption is not acceptable.

Multiple Perimeters

As mentioned above, the design of the GIACE network rests on the concept of "defense-in-depth," therefore multiple perimeters are defined within the network. Perimeters are defined at the border router, the firewall, the internal router, and the

server. This sets up a concentric circle pattern of perimeters which traffic must flow through in order to reach any given device on the network. This layered approach is demonstrated in the diagram below. This is a representative diagram of the GIACE network; individual systems have be eliminated to reduce complexity:



The most external perimeter exists at the edge of the network, with the border routers forming the perimeter of the GIACE network. This perimeter primarily serves to weed out generally undesirable traffic, such as DoS (when possible), spoofed IP addresses, or other high-volume "noise" such as MS-NetBIOS traffic. (A high volume of MS-NetBIOS traffic is currently transiting the Internet due to the release of a variety of worms exploiting flaws in the Microsoft line of products.) The border routers should not be used to replace or duplicate firewall functions, but to improve the efficiency and effectiveness of the firewall and IDS systems.

The next perimeters are defined by the firewall, forming the DMZ and Internal segments. This perimeter is the primary traffic control between the Internet, the DMZ and the Internal network. Since this perimeter will have the highest occurrence of hostile traffic and the highest traffic control requirement, this is where the majority of traffic control should be performed. Logging will occur at the highest levels here. Physical connections will define these perimeters in order to eliminate any possibility of administrative error or circumvention of logical definitions.

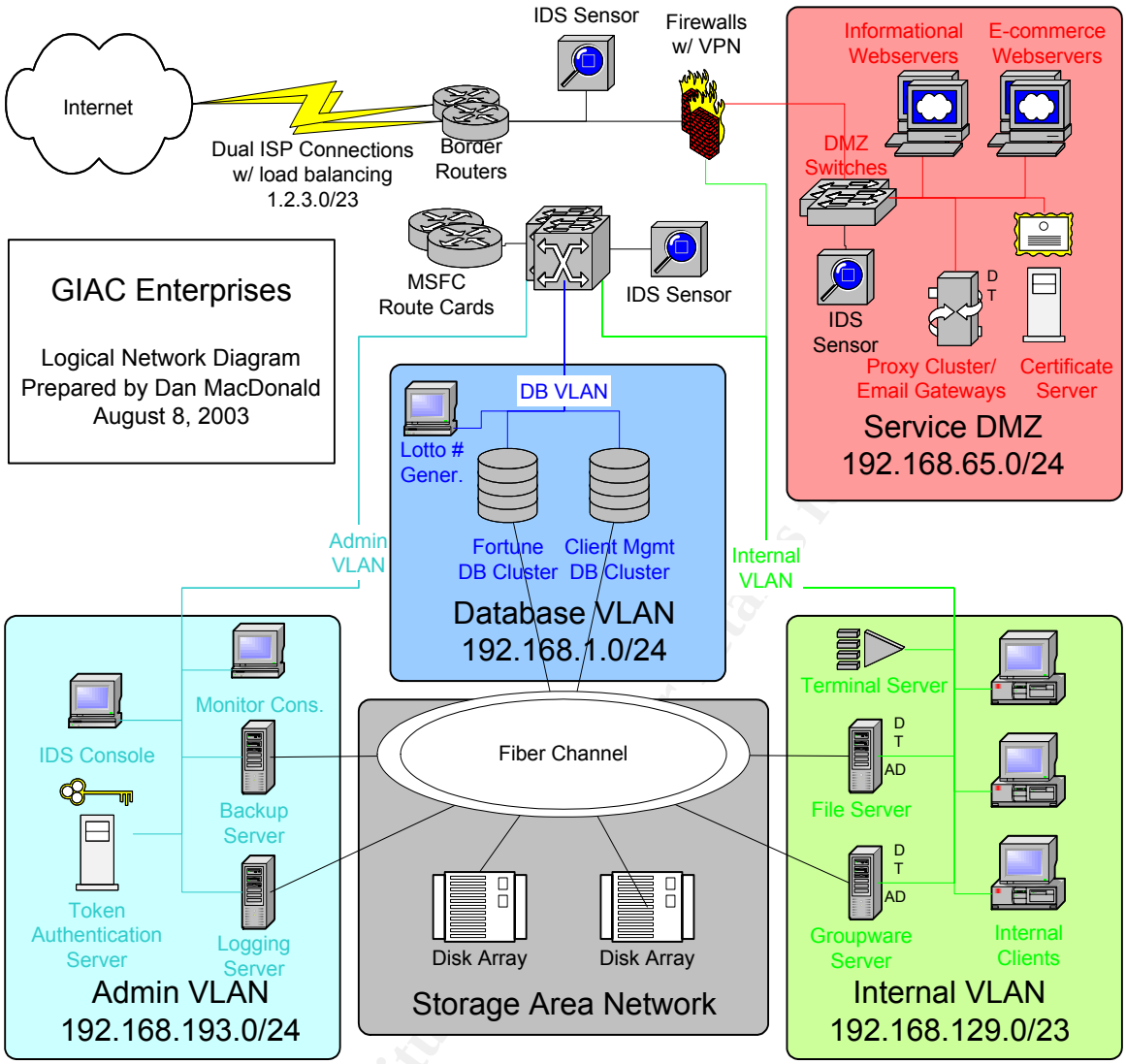
Each system on the DMZ segment will contain a “local” perimeter in the form of host firewall (also known as personal firewall) rule definitions on each server. This will ensure that each server sends and receives only acceptable traffic flows, regardless of the operation of other security devices. Due to the extra management overhead, the local perimeter definition will not be carried throughout the Internal segment at this time.

The Internal routers will perform traffic filtering function in order to define a perimeter for each VLAN on the Internal segment. Traffic filtering responsibilities between the local segments are expected to be relatively light in relation to traffic forwarding responsibilities. This leads to the choice of filtering router equipment over additional firewalls in order to provide greater performance.

The multiple perimeter approach used here provides “defense-in-depth” over the GIACE network, providing three control points for traffic traversing the firewalls, and at least two control points for all traffic originating or terminating on the DMZ segment. Only traffic originating and terminating on the same Internal VLAN may traverse any part of the GIACE network without passing a control point.

Network Architecture

The network architecture looks like this:



As shown in the above diagram, the network will consist of four LAN segments and one SAN segment inside the firewall: one physically disparate segment, three segments logically defined by VLAN, and one fiber channel SAN.

The service DMZ segment will be physically separated for two reasons. First, physical separation will limit the possibility of circumventing the VLAN segmentation either by flaws in the switching logic[1] or by administrative error. Second, this will help to facilitate configuring this segment to allow only defined host-to-host communication within this segment. Through switch and host configuration techniques, hosts on this segment will only be able to communicate with other hosts if explicitly defined to prevent session hijacking attacks.

[1] For examples of possible situations that may result in VLAN hopping, see "Configuration Examples related to VLAN Features" by Cisco Systems, available at http://www.cisco.com/univercd/cc/td/doc/product/lan/28201900/1928v8x/eescg8x/aleakyv.htm. Other techniques exist as well.

The internal VLAN segment will contain all the general-purpose operational servers and workstations. No special measures will be taken on this segment beyond good management practices such as regular patching, centralized logging, etc.

The database VLAN will contain database server clusters containing the business-proprietary fortune and client management databases. The only traffic allowed into or out of this segment will be connections to the database services. The database services will contain extensive security features as well, including multiple permission levels, user authentication, and data normalization. However, the configuration of these features are beyond the scope of this document.

The administrative VLAN will contain systems responsible for the management and security of all systems attached to the GIACE network. All access to this segment from outside will be through appropriately encrypted and/or signed IPSec packets to provide privacy and/or authentication of the packet data.

A Fiber Channel storage array will be physically common to systems on the network with large storage requirements; however, space within the array will be logically portioned so that no system can see another system's storage space. The only exception to this restriction is the backup server, which will require visibility into all storage to properly perform the backup process. This will provide only raw file content (file-level) backup, however the groupware and database servers will require additional clients for record-level backup. The record-level backups for these services will occur through the normal LAN segments.

IP addressing shall consist of private addresses for all devices behind the firewall. Public IP address space has not been assigned from the ISPs as of the time of writing, so the address space 1.2.2.0/23 shall be used for purposes of this design. This may consist of one contiguous block if the space is assigned directly from ARIN, or it may consist of two non-contiguous but portable blocks if assigned by the ISPs selected. Due to the estimated number of personnel at GIACE being approximately 150 employees, large or complex private IP addressing schemes are not expected. Providing maximum flexibility and scalability is one of the stated goals of this design, therefore IP address space is allocated to provide the ability to easily expand the address space using CIDR subnet adjustments if it should be required in the future. The assigned address space is as follows:

Subnet	IP Space	Mask
Service DMZ	192.168.65.0	/24
Database VLAN	192.168.1.0	/24
Internal VLAN	192.168.129.0	/23
Administrative VLAN	192.168.193.0	/24
External	1.2.2.0	/23

The Internal VLAN contains an IP space that is already expanded. GIACE does not expect to realistically consume this many IP addresses on a single broadcast domain, however having extra space within the IP address space allows for administrative flexibility in IP assignments.

Vendor Selection

The vendors selected to provide solutions or platforms in the GAICE network will have a large impact on the traffic flows and protocols used throughout the network. Therefore, vendors will be specified before going further:

- **Switches/Routers:** Cisco Systems, Inc will provide switches and routers. Cisco provides a reasonable price/performance ratio and places a significant amount of emphasis on security features. Specifically, the Cisco 3640 series of routers will be used for border routers, the Cisco Catalyst 3550 switch line will be used in the service DMZ segment, the Cisco Catalyst 6500 series will be used for the core infrastructure, and a Multilayer Switching Feature Card with the Access Control Lists will provide internal routing and traffic filtering.
    - **Border Routers:** The Cisco 3640 router is a medium-performance modular router with high-availability features (HSRP) and scalability to a multiple T1 per router configuration. The 3640 routers will provide advanced traffic optimization through BGP. The border routers will also perform some limited traffic filtering functions to remove some of the 'noise' from the Internet traffic before it arrives at the firewall, and mitigating against spoofing attacks. Routers will run the most current version of the Cisco IOS IP feature set, currently 12.3. Configuration details will be provided in the Security Policy section.
    - **Service DMZ Switching:** The Cisco Catalyst 3550 is the most appropriate switch line for the physically separate service DMZ. The 3550 provides security features such as port security without the higher price tag of a modular switch (such as the Catalyst 4500 series). Catalyst 3550 Switches will run the most current IOS software available, currently 12.1.
    - **Internal Routing/Switching:** High throughput on the internal segments will be maintained by configuring IP Multi-Layer Switching (MLS) between the internal routers and the 6500 switches. Details of the filtering functions will be included in the Security Policy section. The Catalyst 6500 series will run the most recent IOS version for the 6500, 12.2.
- **Firewalls:** Cisco will also provide the firewall solution in the form of a PIX 525 firewall appliance. An appliance firewall was chosen over the blade format available for the 6500 switching chassis due to the desire to have physical separation of the external, service DMZ and internal environments. The PIX 525 provides high-availability and limited IDS features as well. Details of the PIX configuration will be included in the Security Policy section. PIX-OS 6.3 is the most recent available.
- **NIDS:** The open-source Snort package and the IDS features contained within the PIX firewall will provide multiple layers of NIDS monitoring.

- Snort is chosen due to its superior flexibility and price/performance ratio. Snort sensors will be located on mirror ports in the DMZ segment to monitor all servers. Snort sensors will also be located on mirror ports on the internal switch, however only key systems will be monitored in order to prevent oversubscription of the monitor ports. Key servers include: Active Directory (AD) controllers, Terminal Servers, all database servers, and the token authentication server. The IDS console, logging server, backup server (excluding backup dataflow) and monitoring console shall all require IPSec, therefore NIDS monitoring of these devices has minimal value.
- ACID will be used in conjunction with Snort for data aggregation and management. Snort will log to and ACID will read from several platforms of database server; however, the GIACE implementation will utilize MySQL for logging to save expense.[1]
- The PIX IDS features are utilized because they are included with recent versions of the PIX software. The PIX will be configured to only alert on security events. Automatic blocking of traffic shall be used sparingly in order to avoid possible Denial-of-Service (DoS) conditions.

- **Web Servers:** Web servers in the service DMZ shall be Linux servers running the Linux-HA high-availability package and the Apache web server. This selection is primarily due to the better code review process, a highly flexible internal firewall, and superior price/performance ratio as opposed to proprietary solutions such as Microsoft or Sun. All compilers shall be removed from these machines, and all services except Apache shall be disabled or removed from the system. SSH may be enabled on a case-by-case basis from internal workstations only for maintenance and file transfer purposes, however it shall generally be disabled when not in use. FreeS/WAN will provide IPSec negotiation services for communication with systems requiring IPSec. Internal security features such as ext3 journaling, and ACL extensions shall also be used; however, full configuration of these features is beyond the scope of this document. The firewall features of the Linux kernel will also be used, and sample configurations for this feature will be provided in the Security Policy section.
- **Internet Proxy/Email Gateway:** This cluster shall be composed of Linux servers running Trend Micro Interscan VirusWall. This integrated gateway is chosen for its superior feature set and will provide email, web and ftp anti-virus functions, spam control, and proxy services. As with the web servers, all compilers shall be removed, and all services except VirusWall shall be disabled or removed from the system. Appropriate internal security features will be used and a kernel firewall configuration will be provided in the Security Policy section.
- **Certificate Servers:** Certificate Servers shall run Microsoft Windows 2000 with Internet Information Server (IIS) 6 and Microsoft Certificate Authority (CA). Two servers will exist: one in the DMZ for assignment of certificates to relevant external

[1] Further information about ACID is available at http://www.andrew.cmu.edu/~rdanyliw/snort/snortacid.html.

parties, and one on the Administrative VLAN shared with the Authenex authentication server. The DMZ CA shall be the only server in the DMZ running Microsoft software due to Microsoft's expense and poor security track record, however, the CA provided with Windows 2000 allows for relatively easy and inexpensive Public Key Encryption (PKI) management not obtainable elsewhere. GIACE will mitigate some of the risks of IIS by installing URLScan on the IIS server and tightly configuring it to intercept as many improper requests as possible. Due to the sensitive nature of certificate storage, the DMZ CA server will **NOT** contain the root certificate for the PKI structure; instead this certificate will be stored on the internal CA and the DMZ CA shall be a delegate sub-authority.

- **File/Print Servers:** File and print servers shall be Windows 2000 servers running Active Directory (AD). The benefits of having a directory-based security architecture will be realized by GIACE; the expense and extra effort required for Novell E-Directory is not justified in this environment.
- **Terminal Services:** Terminal services shall be Microsoft Windows Terminal Server with Citrix MetaFrame. MetaFrame is required to provide the granular application level control and extended support features desired for GIACE.
- **Groupware:** The groupware server will run Microsoft Exchange 2000. Web mail services will be provided only after authenticating to the VPN. Exchange integrates well with AD and provides a robust feature set and a reasonable price/performance ratio.
- **Workstations:** Workstations will run Windows XP Professional Edition with file and print sharing removed. XP integrates easily into the AD environment and benefits greatly from the manageability and security enforcement features offered by the AD Group Policy.
- **Database:** The database clusters will be Windows 2000 servers running MS-SQL Server 2000 in cluster mode. The scalability and performance of MS-SQL is not as great as that of Oracle or other database servers, however the extra cost is not justified at this point. MS-SQL also integrates well into the AD environment, providing better integration of security features.
- **Authentication:** The token authentication system will be the ASAS authentication system from Authenex. This system consists of an USB token (the A-Key) and client software and will be used for authenticating remote users to the VPN. The A-Key also has certificate storage functionality, so users with assigned A-Key's may also use them for authenticating to certificate-based web services. Other services to leverage the A-Key device are also available and may be used at GIACE in the future. ASAS currently runs on Windows 2000, with support for Linux and other platforms on the near horizon. Final platform choice will be conducted at implementation time based on software evaluation.
- **Logging:** The logging system shall be selected by GIACE closer to implementation, but shall run on a Linux or Windows platform and must accept logging from Syslog.

The security policy provided here is a technical policy; that is, it contains the technical configuration of relevant devices and/or instructions on how to configure the devices to meet the goals outlined above. Configurations provided here are generally representative of one device in each class of devices, as it would be redundant to provide configurations for each identically configured device in a cluster or failover group.

Border Routers

The border routers shall be configured in a redundant fashion using routing protocols and HSRP for rapid failover. A full sample configuration of a border router is included in Appendix B. Included in this section are explanations of the key points of the configuration and how they relate to securing the first perimeter of the GIACE network. Each group of commands, for the most part, may be presented in any order; however, commands presented together in a group may be order dependent. Lines beginning with an exclamation point (!) are comment lines which will not appear after the configuration is loaded to the router; therefore they should be considered documentation in the offline version of the configuration only. Commands which reinforce default settings also will not appear in the online router configurations, so GIACE should not be concerned if not all commands here are listed in a "show configuration" or "show running-config" output.

"service" Commands

Cisco router IOS supports several services that modify both the way the router performs internally and services available to remote hosts on the network. These are controlled through a series of commands beginning with "service".

Congestion is a common occurrence on WAN links. The Nagle algorithm helps to lessen congestion from small TCP/IP packets by holding small packets and combining them while waiting for acknowledgement of previously transmitted packets. It is off by default on Cisco routers; however, it may be useful here, and so is activated as follows:

```
service nagle
```

Packet Assembler/Disassemblers (PADs) are considered legacy equipment, and aren't used at GIACE, so support for them is disabled by the following:

```
no service pad
```

Cisco does not provide much timestamp information by default when generating log entries. The following commands add copious timestamping information to log entries:

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

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

The following command forces all passwords in the router configuration to be encrypted, not just the enable secret:

```
service password-encryption
```

These commands ensure that the routers will not respond to DHCP or BOOTP requests, nor relay them across the router:

```
no service dhcp
no ip bootp server
```

For some reason, Cisco has a pressing need to advertise information all over any locally connected segment. This is both a security and a performance issue, so it shall be disabled:

```
no cdp run
```

"Finger" allows a user to view users logged into a remote system. This is inappropriate for a router, and therefore disabled with the following:

```
no ip finger
```

Cisco routers by default interpret every command that is not understood by IOS as a hostname, causing the router to attempt a DNS lookup. Since most cases are the result of a mistyped command, this has the effect of driving the administrator insane while waiting for IOS to come back from failed lookups. To save administrator sanity, this command disables DNS lookups for hosts:

```
no ip domain-lookup
```

Cisco eventually surrendered to the popular point-and-click movement and added a web configuration interface. The interface contains many bugs and problems, plus all the issues inherent to web management utilities. The following line disables it:

```
no ip http server
```

Logging & SNMP Configuration Commands

Time configuration is only relevant on Cisco routers as it applies to logging. Time synchronization is important for correlation of events across network nodes. The following commands configure Network Time Protocol (NTP) to synchronize to the DMZ proxy servers, which also run NTP. Authentication is configured to prevent any possible spoofing to skew time on the routers. The NTP configuration looks like this:

```
ntp authentication-key 1 md5 09107D2C3A3732263427211375 7
ntp authenticate
```

```
ntp server 1.2.3.20
ntp server 1.2.3.21 prefer
```

While multi-location companies often set all times to a common time zone such as GMT, GIACE expects to remain a single-location organization at this time. Consequently, we will choose the local time zone for convenience:

```
clock time zone EST -5
```

Cisco provides a small logging buffer within the router device. Since the default size of this buffer varies by platform, explicitly defining the size is a good idea. Adjusting the buffer may be necessary to ensure that an appropriate amount of logging is stored on the device. This logging should not be relied upon and should be used only for troubleshooting convenience. The final parameter defines the minimum level message logged to the buffer.

```
logging buffered 16384 notifications
```

Cisco will provide log information to the console when asked. This can be useful, but can also be annoying. The following command disables logging messages on the console, however individual administrators may wish to activate it depending on the task at hand:

```
no logging console
```

We want to log as much as we can to an external server for easier management, better performance, and better correlation of events across the network. The following commands tell the router to log to an external syslog server, tag the log entries with locally defined facility number 5, and specify the server to log to:

```
logging trap debugging
logging facility local5
logging host 1.2.3.2
```

The IP address specified to receive the logs is an IP address assigned to the firewall, which will appropriately NAT the messages to the logging server.

SNMP will be used on the GIACE network for network monitoring and reporting. The following commands define the SNMP engine, set the community string, and define what SNMP traps should be sent, and what server they should be sent to:

```
snmp-server engineID local 00000009020000216647FB5
snmp-server community GIACE-strong-snmp RO 20
snmp-server enable traps snmp
snmp-server enable traps hsrp
snmp-server enable traps config
snmp-server enable traps syslog
snmp-server host 1.2.3.2 bdr-routers
```

Device Access Configuration Commands

Strong passwords should be used everywhere, and the border routers are no exception. The following command sets the password for accessing “enable” mode, or gain administrative level access to the router.

```
enable secret cisco
```

Obviously this password is not sufficiently strong, and an alternate should be chosen. After the issue of this command, the router will encrypt the password and store it in the configuration in hash form.

Cisco does not define individual user accounts by default, however the capability does exist within IOS. Maintaining individual user accounts on network equipment is unnecessarily inconvenient in most environments, and the GIACE environment is not large enough to justify running a TACACS server to centralize access to network equipment. Therefore, a single account will be defined simply to ensure that any hacker will have to guess the username as well if a hacker should gain access to a router console:

```
username GIACEuser password cisco
```

The router will show a warning banner to all users connecting to the console. Warning banners sometimes make prosecution easier in case of an incident. The banner is defined by the following:

```
banner motd #

Router GIACE-bdr-1.  Access to this device or attached networks by
unauthorized users is prohibited.  No user of this device or connected
networks has any expectation of privacy.
Violators will be prosecuted.

#
```

Cisco supports a variety of methods for connecting to a router Command Line Interface (CLI). Most Cisco platforms (including the 3640 routers used by GIACE) contain a console port, an auxiliary port, and will accept connections in-band via telnet, RSH, or SSH if configured to do so. As mentioned above, GIACE will not allow in-band management of any network device; instead management must be performed through a console connection accessed through a terminal server. Therefore, only the console port is left active. The following commands force users to login when connecting to the console port and disable the auxiliary port and all virtual terminals (VTYs):

```
line con 0
 login local
 transport input none
line aux 0
 transport input none
 transport output none
```

```
line vty 0 4
 no login
 transport input none
 transport output none
```

Interface Commands

Each router interface must be individually configured before it will pass traffic. This section will detail each command used for this purpose and state what interfaces the command applies to. The initial design for GIACE specifies two T1 lines, one connected to each router. GIACE will likely need to add more T1 lines in the future to account for growth, therefore configuration for any serial interface may be applied to interfaces added in the future to facilitate more communications lines.

Two special pseudo-interfaces are provided by Cisco IOS: the null interface and the loop back interface. The null interface discards all packets routed to it, and must be configured on each router in order to accommodate black-hole routes (described below). Also, the routers should not generate ICMP unreachable packets for any traffic that is routed to the interface. Both are accomplished with the following commands:

```
interface Null0
 no ip unreachables
```

The loop back interface exists to provide a method of ensuring that at least one IP address on the router can never change or be driven offline. This can be important for some routing protocols such as OSPF; however, GIACE is not using a complex routing protocol at the border. Therefore, the loop back interface is left unconfigured at this time.

Each interface may be designated with a user-defined description that serves only to provide self-documentation within the configuration. GIACE will use this description on each router interface for clarity. Here is an example of such a description:

```
description T1 to ISP
```

Since this is an Internet router, each interface must have an IP address configured. The following command explicitly defines the IP address for an Ethernet interface:

```
ip address <ip_address> <netmask>
```

Serial interfaces may be configured in point-to-point mode (i.e. PPP) however, this can be wasteful of IP addresses. Cisco supports IP sharing with other interfaces in this case. The following command configures the interface to use the IP address assigned to interface FastEthernet0:

```
ip unnumbered FastEthernet0
```

GIACE will filter out certain traffic based on access lists (ACLs) defined in the ACL section. The following commands will assign the ingress filtering ACL to traffic entering the serial interface, and the egress filtering ACL to traffic entering the Ethernet interface, respectively:

```
ip access-group antispoof-ingress in
ip access-group antispoof-egress in
```

Each interface will also filter against propagating certain noisy traffic as defined in the anti-noise list. This list is applied to each interface with the following:

```
ip access-group antinoise out
```

The following commands suppress ICMP Redirect, and ICMP Unreachable messages:

```
no ip redirects
no ip unreachables
no ip mask-reply
```

ICMP Redirect suppression on the Ethernet interfaces will cause all traffic to pass through the active HSRP router, even if the standby router has a more efficient route. In this case, the active router will directly route the traffic to the passive router which will then route it normally. ICMP Redirect packets are useful to tell the client to route future packets to that address to the correct router however, lessening the load on the active router. Therefore, ICMP Redirect messages should not be suppressed on the Ethernet interfaces to make load sharing between the routers much more efficient.

Proxy ARP allows IP clients that do not understand a subnet mask to route properly when IP address spaces are subnetted. Modern IP stacks understand subnet masks, therefore proxy ARP should almost always be disabled with the following command:

```
no ip proxy-arp
```

The following command is applied to the Ethernet interfaces and prevents ARP table entries from timing out:

```
arp timeout 0
```

This makes it near impossible to perform ARP poisoning attacks against the router, as the switch will never forget ARP entries. (The routers should only communicate to each other and the PIX firewalls from the Ethernet interfaces.) A permanent ARP cache also gives the benefit of recording all hosts on the local segment that have ever attempted to talk to the switch, such as by a ping sweep, port scan, or telnet attempt. This has very limited benefits if not checked regularly, but may be useful if questions arise about the behavior of machines in the DMZ.

Suppress directed broadcasts so GIACE doesn't become a traffic amplifier:

```
no directed-broadcast
```

The following command configures the router to keep count of access list violations:

```
ip accounting access-violations
```

This command keeps multicast traffic from crossing the router. We don't want to limit multicast by address because it is used for some routing protocols and HSRP:

```
ip multicast boundary 30
```

Rate limiting helps to provide resistance to Denial-of-Service (DoS) attacks. The first command limits traffic defined in access-list 150 to 2Mbps, dropping traffic that exceeds 200Kb/s. The second command limits traffic defined in access-list 160 to approximately 50Kb/s, dropping the excess. Access-list 150 defines UDP traffic, 160 defines ICMP:

```
rate-limit input access-group 150 200000 25000 25000 conform-action transmit
      exceed-action drop
rate-limit input access-group 160 50000 6250 6250 conform-action transmit
      exceed-action drop
```

This command activates Cisco NetFlow switching. NetFlow switching allows greater routing efficiency by maintaining a state table, therefore allowing certain packets to bypass ACL lists and other steps in the process without compromising security. NetFlow also allows tracking of certain statistics, although these features will not be used by GIACE initially. The following command activates NetFlow:

```
ip route-cache flow
```

Suppress CDP on each interface, just in case it ever gets activated router-wide:

```
no cdp enable
```

With the combination of the above commands, a serial interface will be configured similar to this:

```
interface Serial0/0
 description T1 to ISP
 ip unnumbered FastEthernet0
 ip access-group antispoof-ingress in
 no ip redirects
 no ip unreachables
 no ip proxy-arp
 no ip directed-broadcast
 no ip mask-reply
 ip accounting access-violations
 ip multicast boundary 30
 rate-limit input access-group 150 200000 25000 25000 conform-action transmit
      exceed-action drop
 rate-limit input access-group 160 50000 6250 6250 conform-action transmit
      exceed-action drop
```

```
 ip route-cache flow
 no shutdown
 no cdp enable
```

The Ethernet interfaces have one additional configuration component: Hot Standby Router Protocol (HSRP). HSRP allows for rapid (a few seconds) transfer of response to a virtual IP and MAC address between routers in the event of router or line failure. The following commands configure HSRP:

```
standby authentication GIACE-ro
standby name GIACE-bdr-gw
standby ip 1.2.3.1
standby track Serial0
```

The first command defines an identifier for the HSRP group to use to ensure that routers do not respond to the wrong HSRP group accidentally. The second names the group. The third defines the virtual IP address shared among the group. (This is the address that should be configured as the gateway for all systems using the router group.) The last command allows the router to track a particular interface and attempt to remove itself as the active router if the defined interface goes down or fails. Implementers should note that this configuration is for later versions of the 12.3 IOS software, so care should be taken to reconfigure if necessary.

With the addition of the HSRP configuration, an Ethernet interface configuration will be similar to the following:

```
interface FastEthernet0/0
 description Untrusted Segment Interface
 ip address 1.2.3.253 255.255.254.0
 ip access-group antispoof-egress in
 no ip redirects
 no ip unreachables
 no ip proxy-arp
 no ip directed-broadcast
 no ip mask-reply
 ip accounting access-violations
 ip multicast boundary 30
 ip route-cache flow
 speed auto
 no shutdown
 no cdp enable
 standby authentication GIACE-ro
 standby name GIACE-bdr-gw
 standby ip 1.2.3.1
 standby track Serial0/0
```

IP & Routing Configuration

The commands described in this section affect how the router treats certain IP features and define certain routing characteristics. Normally, configuration of the chosen routing protocol would be covered here also; however, the choice of routing protocol(s) for

these routers will depend greatly on the choice of ISP. BGP is a likely choice, however BGP configuration will be highly ISP dependent and is beyond the scope of this document. When the routing protocol is chosen and configured, care must be taken to guard against route spoofing and injection for these routers.

Other configuration includes enabling the full IP range:

```
ip subnet-zero
```

Source-routed packets have preferred routing information specified within the packet in an attempt to override the route that a packet would normally travel. Source routing has no legitimate use on the Internet today. This command disables the handling of source-routed packets:

```
no ip source-route
```

Enabling classless routing allows for more efficient route summarization:

```
ip classless
```

This is a default route definition. The following line should be changed to reflect correct information received from GIACE's ISP:

```
ip route 0.0.0.0 0.0.0.0 1.1.1.1
```

Directing RFC-1918, unassigned, or otherwise invalid addresses to the Null interface will cut down on the possibility of spoofed addresses entering the GIACE network. The following route statements are abbreviated here; however, the full set of null routes is included in the sample border router configuration in Appendix B. In the future, these routes may be aggregated to save memory if necessary:

```
ip route 2.0.0.0 255.0.0.0 Null0
ip route 5.0.0.0 255.0.0.0 Null0
ip route 7.0.0.0 255.0.0.0 Null0
ip route 10.0.0.0 255.0.0.0 Null0
.
.
.
ip route 127.0.0.0 255.0.0.0 Null0
ip route 169.254.0.0 255.255.0.0 Null0
ip route 172.16.0.0 255.240.0.0 Null0
.
.
.
ip route 192.0.2.0 255.255.255.0 Null0
ip route 192.168.0.0 255.255.0.0 Null0
ip route 197.0.0.0 255.0.0.0 Null0
ip route 223.0.0.0 255.0.0.0 Null0
```

Access Control Lists (ACLs)

This section covers the heart of the router configuration from a security point of view. These ACLs define the first perimeter control to define what traffic is allowed to cross to the GIACE network from the Internet and vice versa. Each list starts with a "no access-list" command to ensure the list is clear before defining it; this command will not appear in the final configuration.

Egress Anti-spoofing ACL

This list suppresses outbound traffic to eliminate any possibility of address spoofing from the GIACE network. Any traffic claiming to originate from an address that isn't assigned to GIACE will not be allowed to leave the GIACE network:

```
no ip access-list extended antispoof-egress
ip access-list extended antispoof-egress
 remark Anti-Spoofing - egress traffic
 remark Allow only traffic coming from an address assigned to GIACE
 permit ip 1.2.2.0 0.0.1.255 any
 deny   ip any any log-input
```

Ingress Anti-Spoofing ACL

At first this ACL is similar to the Egress ACL, except that it is applied to traffic from the Internet. Therefore, the address space must be more carefully defined. Note that the 1.0.0.0/8 block is denied, even though the external address space GIACE is currently using falls within this block. This is because we are filtering based on source address, therefore we should never see packets coming *from* the GIACE address block on the Internet side of the border routers. As with the null route list above, this list is abbreviated, but is included in full in Appendix B.

```
no ip access-list extended antispoof-ingress
ip access-list extended antispoof-ingress
 remark Anti-Spoof (Incoming)
 remark Knock down RFC-1918, unassigned, and otherwise undesirables on the
      way into the network
 remark Blocks are aggregated to help speed ACL processing.
 deny   ip 1.0.0.0 0.255.255.255 any log-input
 deny   ip 2.0.0.0 0.255.255.255 any log-input
 deny   ip 5.0.0.0 0.255.255.255 any log-input
 deny   ip 7.0.0.0 0.255.255.255 any log-input
 deny   ip 10.0.0.0 0.255.255.255 any log-input
 .
 .
 .
 deny   ip 169.254.0.0 0.0.255.255 any log-input
 deny   ip 172.16.0.0 0.15.255.255 any log-input
 .
 .
 .
 deny   ip 192.0.2.0 0.0.0.255 any log-input
 deny   ip 192.168.0.0 0.0.255.255 any log-input
```

```
deny   ip 197.0.0.0 0.255.255.255 any log-input
deny   ip 223.0.0.0 0.255.255.255 any log-input
deny   ip 224.0.0.0 31.255.255.255 any log-input
```

Another function of this ACL is to clean up some of the Internet "noise." Most of this occurs on another ACL, however, this is the first ACL that packets coming from the Internet will traverse, and therefore this list contains filtering for some packets that should absolutely not go any further, not even to the router.

First, kill fragmented ICMP. Fragmented ICMP should not occur under normal circumstances, and therefore is likely to be malicious. We don't want any possibility of it even exploiting vulnerability on the router:

```
remark Kill ICMP fragments - should never be necessary to frag ICMP
deny   icmp any any log-input fragments
```

Also, we want to kill multicast at the earliest opportunity to guard against any possibility of someone injecting bad traffic into the routers or interfering with HSRP:

```
remark GIACE's multicast usage is local only
deny   ip any 224.0.0.0 15.255.255.255
remark Let everything else through, at least to the firewall
permit ip any any log-input
```

Anti-Noise

This ACL serves to cut some of the Internet "noise" before it reaches the firewall and prevent leakage of traffic from the GIACE network to the Internet in case of a misconfigured firewall or other circumstances. The traffic blocked by this list is primarily directed by high-risk services not used by GIACE on the Internet, such as NetBIOS, X-Windows, IRC, etc. For ease of maintenance, this "noise" ACL should be separate from the anti-spoofing ACLs defined above. Also, a separate ACL will allow the same ACL to be applied to traffic crossing the router in both directions. However, Cisco only allows one ACL per direction per interface, and since the above "spoofing" ACLs use the inbound ACL definition for each interface, we are left applying this ACL to the outbound on each interface. This potentially could provide risk to the router, if it were running any external services. Since we have already disabled all services on the router, applying the ACL outbound should not provide a problem in the case.

GIACE will periodically maintain this ACL for problematic traffic. Entries on the "SANS Top 20 Vulnerabilities" list are good candidates for blocking by this ACL, as are ports that have high volumes of worm activity. Careful sanity checking of each entry is necessary however. For example, the "SANS Top 20" recommends blocking ports tcp/80 (HTTP) and ports tcp/443 (HTTPS). Blocking these ports will render GIACE completely inoperable since GIACE is a web-based E-Commerce company!

The purpose of blocking here is not to duplicate the functionality of the firewall cluster, but to remove some of the high-volume problem traffic, eliminate traffic that may be a

risk before reaching the firewalls or a risk to the firewalls directly, and to provide an extra line of defense against a misconfigured firewall or operator error.

Ports selected for blocking at the border router are as follows:

Ports	Service	Reason
tcp/1-19 udp/1-19	Small Services	No significant use for these ports, however they may provide traffic amplification abilities for DoS attacks.
tcp/23	Telnet	Guard against administrative error if Telnet is accidentally enabled.
tcp/67-68 udp/67-68	DHCP	No reason to cross border router. May leak network information.
udp/69	TFTP	May be used for backchannel after breach.
tcp, udp/135 tcp/137-139 tcp, udp/445	NetBIOS	Large exposure on these overloaded ports – authentication, file sharing, RPC etc. These ports are also target for a large amount of worm activity.
tcp/161-162 udp/161-162	SNMP	Information leakage – will have to add an exception for allowing traps to leave the router however.
tcp/1433 udp/1434	MS-SQL	Large amount of worm traffic still from MS-Slammer in January 2003.
tcp, udp/194 tcp, udp/994 tcp, udp/6667 tcp/7666	IRC	IRC is often used as a worm back channel.

The rules for this ACL should be tuned for performance according to current traffic flows. In the meantime, this ACL provides a reasonable guess from which to start:

```
no ip access-list extended antinoise
ip access-list extended antinoise
 remark Anti-Noise ACL
 remark Knock down noisy and undesired traffic at the earliest opportunity
 remark NetBIOS provides the largest amount of noise
 deny   tcp any any range 137 139 log-input
 deny   tcp any any eq 135 log-input
 deny   udp any any eq 135 log-input
 deny   tcp any any eq 445 log-input
 deny   udp any any eq 445 log-input
 remark MS-SQL also has a lot of traffic
 deny   tcp any any eq 1433 log-input
 deny   udp any any eq 1434 log-input
 remark TCP and UDP small services
 deny   tcp any any range 1 19 log-input
 deny   udp any any range 1 19 log-input
 remark DHCP
 deny   tcp any any range 67 68 log-input
 deny   udp any any range bootps bootpc log-input
 remark SNMP - must permit from GIACE addresses
 permit tcp any 1.2.2.0 0.0.1.255 range 161 162 log-input
 permit udp any 1.2.2.0 0.0.1.255 range snmp snmptrap log-input
```

```
deny   tcp any any range 161 162 log-input
deny   udp any any range snmp snmptrap log-input
remark TFTP is often used as a backchannel following a breach
deny   udp any any eq tftp log-input
remark IRC is often used as a worm backchannel/control channel
deny   tcp any any eq irc log-input
deny   udp any any eq 194 log-input
deny   tcp any any eq 994 log-input
deny   udp any any eq 994 log-input
deny   tcp any any eq 6667 log-input
deny   udp any any eq 6667 log-input
deny   tcp any any eq 7666 log-input
remark Block telnet for added safety
deny   tcp any any eq telnet log-input
remark Let everything else through
permit ip any any
```

Note that ICMP is not restricted in this ACL. Traceroute is already ineffective as all interfaces are configured not to send information leaking ICMP messages such as unreachable, and the firewalls will likewise not send any useful ICMP messages. ICMP restrictions may be added to this ACL in the future to limit ping or other ICMP features.

Electing not to limit ICMP might leave the routers vulnerable to issues such as backchannel data with Loki. GIACE will rely on the IDS and firewalls for alerting and protection against these types of occurrences, and elects to not impede the designed purpose of ICMP further than is already done through the device configurations.

SNMP

The following ACL defines allowed access to the SNMP services configured on the border routers:

```
no access-list 20
access-list 20 remark SNMP controls
access-list 20 permit 1.2.3.2
access-list 20 permit 1.2.3.250
access-list 20 permit 1.2.3.251
access-list 20 deny   any log
```

Multicast

The following ACL defines all multicast addresses in order to suppress multicast from crossing the border routers:

```
no access-list 30
access-list 30 remark Multicast boundary definition
access-list 30 deny   224.0.0.0 0.0.16.255
```

DoS Prevention Lists

The following lists define traffic patterns for rate limiting in and attempt to limit DoS attacks. List 150 defines all UDP traffic, and list 160 defines all ICMP traffic:

```
no access-list 150
access-list 150 remark Rate-Limit UDP
access-list 150 permit udp any any

no access-list 160
access-list 160 remark Rate-Limit ICMP
access-list 160 permit icmp any any
```

Firewalls

The Internet firewall pair chosen is the Cisco PIX version 6.3, the latest version available at the time of writing. This section primarily focuses on a description of the commands used in the final configuration. A complete configuration for the active PIX firewall is included in Appendix C.

General Configuration Commands

The following commands are used for the basic configuration of the PIX device. Most of the commands are self-explanatory, so minimal treatment will be given here. Focus instead will be on the more critical commands treated in later sections.

The first command serves to define the interface speed and duplex characteristics. While the PIX detects these settings automatically, it is desirable to explicitly set them to speed up link up/down changes and to insulate against the accidental or deliberate introduction of a hub directly adjacent to the PIX device. (Installation of a hub would enable simple sniffing attacks against all traffic flowing through the PIX and degrade performance. A command to force the first Ethernet device to 100 Mb, full-duplex mode looks like this:

```
interface ethernet0 100full
```

For administrative convenience, the PIX encourages the use of pseudonyms for the individual interfaces. An example of a command to set the name for the first interfac:

```
nameif ethernet0 outside security0
```

The “security0” parameter sets the trust level of hosts accessible through that interface. The level 0 defines the attached network as completely hostile and never to be trusted. Level 100 defines the attached network as completely trusted, and allows the interface to access all lower security levels. Levels in-between trust only networks with lower security levels. For example, an interface at level 10 will trust the “outside” interface defined above, and therefore be able to communicate with any host reachable via that interface, but will not be able to reach hosts via any interface with a level of 10 or

higher. This concept is not used in this firewall configuration because traffic will be limited in both inbound and outbound directions through the use of Access Control Lists (ACLs).

The following command defines the enable password for the PIX device:

```
enable password <password> encrypted
```

Obviously, something stronger than <password> will be used in the final configuration.

The following commands set the hostname and default domain name, respectively, for the PIX:

```
hostname GIACE-fw-a
domain-name giace.com
```

The following defines the time zone the firewall will use to display timestamps. As with the routers above, we will use the local time zone due to the single geographic location:

```
clock time zone EST -5
```

Set the PIX to display 24 lines before pausing for user input:

```
pager lines 24
```

The following command explicitly sets the maximum MTU size for the interface:

```
mtu outside 1500
```

Define the IP addresses for an interface:

```
ip address outside 1.2.3.250 255.255.254.0
```

Define a pool of addresses for assignment to VPN clients:

```
ip local pool remote-vpn 192.168.253.10-193.168.253.59
```

The following commands define how the PIX will handle triggers on the limited IDS rule set supported by the PIX. The internal PIX IDS places alerts in two categories. The first are informational alerts, which consist of activity that is suspicious, but not necessarily malicious. The second category, attack, consists of traffic that has a high probability of being an attack. The first command sets the PIX to send an alert on informational IDS triggers, and the second sets the PIX to reset the connection when attack traffic is seen. The PIX also has support for dropping packets, but reset was chosen to ensure that servers do not suffer a DoS condition, either intentionally or as an inadvertent side effect. The reset will allow the server to remove the connection from the state table immediately, preventing a SYN flood condition. Very little extra information is given to the attacker from the reset, since the attacker already knew

about the availability of the service in most or all cases. The IDS configuration commands follow:

```
ip audit info action alarm
ip audit attack action reset
```

Require the firewall to reassemble all fragments before passing:

```
fragment chain 1 outside
fragment chain 1 inside
fragment chain 1 dmz
```

The following command defines how long an ARP entry should remain in the ARP table:

```
arp timeout 1440
```

This setting is defined in seconds, so the above is for four hours. Unlike with other network devices, a permanent ARP table would be unmanageable due to the number of devices and the use of DHCP on the internal segment.

The following defines the default route:

```
route outside 0.0.0.0 0.0.0.0 1.2.3.1 1
```

State table entries for various classes of traffic are set as follows:

```
timeout xlate 1:00:00
```

The second parameter is the class of traffic. The last parameter is the time in HH:MM:SS format. These timers are reset when traffic matching the entry is seen, so in most cases these timers behave as idle timers. In the example above, the translate (xlate) table is set to one hour, effectively limiting all TCP connections to one hour without traffic. The PIX has many classes of traffic that may be adjusted in this fashion.

Next, disable web, telnet, and SSH management of the PIX. Disabling HTTP also has the effect of disabling the PDM management features:

```
no http server enable
no telnet
no ssh
```

The only SNMP string defined by default on Cisco devices at this point is “public,” however we will issue the following commands to ensure that none are left over from previous configurations:

```
no snmp-server community public
no snmp-server community private
no snmp-server community cisco
no snmp-server community write
```

Further SNMP configuration settings will be discussed in the SNMP Configuration section below.

The following enables flood guard to ensure that users attempting to authenticate to the PIX and not completing their authentication do not create a DoS condition. This condition is not likely in this configuration, since the only user authentication performed by the PIX is for VPN; however, flood guard does not provide much of a performance issue, so it is enabled here:

```
floodguard enable
```

The following command allows VPN clients to bypass ACL checking. This means that all traffic arriving via the IPSec VPN will automatically be accepted and routed to the appropriate interface. While this is administratively convenient, it does remove the ability to filter certain traffic from crossing the VPN, such as IRC or other hostile or undesirable traffic. The command follows:

```
sysopt permit-ipsec
```

NAT Configuration Commands

This section details the command sets used to configure Network Address Translation (NAT) on the PIX firewalls. NAT is used to change addresses from one IP range to make them appear as if they were from another IP range. NAT is most commonly used with RFC-1918 addresses, or addresses that are reserved for private use. NAT offers some limited security-by-obscurity benefits to network by hiding IP address ranges and network topology information from a public network at large, such as the Internet. NAT also allows for better utilization of the shrinking IP address space, as not every device requires a public IP address.

The first group of commands defines address pools of public addresses:

```
global (outside) 65  1.2.3.70-79 netmask 255.255.254.0
global (outside) 193 1.2.3.80-89 netmask 255.255.254.0
```

The parameters (in order) define the interface to apply the group to a group number, the range, and the appropriate netmask. While the groups are not yet assigned to a particular IP block for NAT, the group number is consistent with the third octet of the IP range, so these definitions are for the DMZ (192.168.65.x) and admin (192.168.193.x) segments. The database and internal segments do not talk to the Internet directly and therefore do not require NAT groups.

Multiple address groups are defined here to allow the different network segments to use different groups of public addresses. This weakens the use of NAT for obscuring network topology information, however security-by-obscurity should not be relied upon

as an effective security mechanism anyway. In this case, the added administrative visibility outweighs the benefits of complete obscurity.

The next group of commands applies the NAT groups to a particular IP address range:

```
nat (inside) 193 192.168.193.0 255.255.255.0 0 0
nat (dmz) 65 192.168.65.0 19 0 0
```

Group 193 is assigned to IP addresses assigned to the administrative network, which must talk to the outside directly to receive logging and SNMP information from the border routers. Group 65 is assigned to the DMZ that must talk to the outside to provide services to both internal and external users. The database and internal segments as mentioned above do not get NAT since hosts on these segments must use the proxy servers to access the internet. The final statement disables NAT for VPN users since VPN users also must use the proxy servers to access the Internet.

The final group of commands defines the static translations for incoming traffic:

```
static (dmz, outside) tcp 1.2.3.10 www 192.168.65.10 www netmask
      255.255.255.255
static (dmz, outside) tcp 1.2.3.11 443 192.168.65.11 443 netmask
      255.255.255.255
static (dmz, outside) tcp 1.2.3.20 smtp 192.168.65.20 smtp netmask
      255.255.255.255
static (dmz, outside) tcp 1.2.3.21 smtp 192.168.65.21 smtp netmask
      255.255.255.255
static (dmz, outside) tcp 1.2.3.50 443 192.168.65.50 443 netmask
      255.255.255.255
static (inside, outside) 1.2.3.241 192.168.193.11 netmask 255.255.255.255
static (inside, outside) 1.2.3.242 192.168.193.12 netmask 255.255.255.255
static (inside, outside) 1.2.3.244 192.168.193.14 netmask 255.255.255.255
```

These commands are restricted to specific ports in order to limit the possibility of administrative error accidentally exposing more ports to the Internet on the service hosts than is required. The final three commands are not restricted by port, since they are principally for IPSec traffic. IPSec uses ip types 50 or 51 depending on whether Authentication Header (AH, packet signing) or Encapsulating Security Payload (ESP, packet encryption) is used; however the PIX does not support IP type definitions in "static" statements, only TCP or UDP ports.

"fixup" Commands

The PIX firewall supports stateful inspection of packet payloads and dynamic firewall configuration for protocols that do not have defined fixed ports. Both of these features are controlled through the use of "fixup" commands.

Some protocols are not used or desired at GIACE. Consequently they are disabled with the following statements:

```
no fixup protocol h323 h225
no fixup protocol h323 ras
no fixup protocol ils
no fixup protocol rsh
no fixup protocol sqlnet
no fixup protocol sip
no fixup protocol skinny
```

The remaining protocols are desired and need configuration of one form or another:

```
fixup protocol dns maximum-length 512
fixup protocol ftp 21
fixup protocol http 80
fixup protocol rtsp 554
fixup protocol smtp 25
```

The DNS fixup configures a maximum payload length for DNS traffic. This strips out some buffer overflows, DoS attacks, and attempts to mine the DNS database.

The FTP fixup allows active FTP without requiring opening traffic from port 20. When an FTP session is initiated through the PIX, it is recorded in the state table. During the PIX's inspection of the FTP payloads, the PIX learns the negotiated ports for the data transfer port for the FTP session. The PIX will then dynamically allow traffic on the FTP data ports recorded as part of inspection of legitimate FTP streams passing through the PIX.

The HTTP fixup command enables URL logging. URL screening by third party software and Java/ActiveX filtering are also supported with the HTTP fixup. Neither of the latter features is currently activated, but GIACE may choose to implement them in the future. Similar to the FTP fixup, the RTSP fixup allows the dynamic ports used by streaming traffic such as RealAudio or QuickTime to pass through the firewall if a legitimate connection is recorded in the PIX state table.

The SMTP fixup provides command set limiting and SMTP response modification. Message modification insulates the SMTP server from information leakage by replacing most of the banner and many status message text values with asterisks (*). For the most part, only the numeric part of the message is allowed to pass through the firewall to the outside. Command set limiting restricts the commands available to the seven absolutely necessary commands (HELO, MAIL, RCPT, DATA, RSET, NOOP, and QUIT) from outside the firewall, which also helps limit information leakage in the case of some commands (such as EXPN and VRFY, which will leak legitimate email addresses on the mail server). Command set limiting also takes away certain other, more malicious, uses of other commands and helps to limit the commands available to attack the server.

Logging & SNMP Configuration Commands

The PIX firewalls shall log to the syslog server on the administrative network. The first command activates logging:

```
logging on
```

The next command defines what logging facility to use with the server. Facility 19 translates to local3 on the syslog server, and is what many syslog servers expect to see:

```
logging facility 19
```

Set the buffer to log into memory events of level 4 (warnings) and higher:

```
logging buffered 4
```

We will display logging information on the console as well. Administrators may adjust this, however, as the logging on the console can be distracting when actively maintaining the device. The command follows:

```
logging console 5
```

Define the IP address to receive the logging information:

```
logging host inside 192.168.193.14
```

Send traps to the SNMP host as well for level 5 (notifications) and higher:

```
logging trap 5
```

By default, the PIX will send logging information from the active router in an active-standby pair, so this command will activate logging from the standby router:

```
logging standby
```

The following set the SNMP location and contact values respectively:

```
snmp-server location GIACE
snmp-server contact Security Administrator
```

SNMP uses a “community string” for identification and authentication purposes. This is not a strong security scheme, especially since the string is passed in clear text over the network. Therefore, the community string set in the following command should follow password best practices (long, complex, and unique):

```
snmp-server community <giace_strong_community>
```

The following commands define which host to send SNMP traps to and to accept SNMP requests from:

```
snmp-server host 192.168.193.11 trap
snmp-server host 192.168.193.11
```

And finally, this command enables the sending of traps to the above host:

```
snmp-server enable traps
```

Authentication Server Configuration Commands

GIACE has elected to require token authentication for access to the VPN and has chosen the Authenex solution. This solution comes in the form of a USB key for the client, and a RADIUS server on the server side. The following commands configure the PIX firewall to authenticate against the Authenex server:

```
aaa-server authenex protocol radius
aaa-server authenex host 192.168.193.15 <radius_key>
aaa-server radius-acctport 1813
aaa-server radius-authport 1812
```

The first command defines an AAA server named “authenex” as a radius server. The second command defines the IP of the RADIUS server and the RADIUS key required for authentication. The following two commands specify that the PIX should use the alternate RADIUS ports, instead of the defaults.

The AAA server must still be assigned to a VPN profile; this will be treated in the IPSec/VPN Configuration section.

Failover Commands

The ability to do a stateful failover is one of the reasons for choosing the PIX firewall. The next group of commands configures failover settings on the firewall, including using a dedicated Ethernet interface for standby as opposed to the serial cable required prior to version 6.x. Using a dedicated interface guarantees sufficient bandwidth availability for the state information required to maintain stateful failover.

The first command activates the failover features:

```
failover
```

The next group of commands configures the IP and MAC addresses of the corresponding interface of the standby firewall in the pair:

```
failover ip address outside 1.2.3.251
failover ip address inside 192.168.129.253
failover ip address dmz 192.168.65.253
```

```
failover ip address failover 192.168.254.2
failover mac address outside <act_mac> <standby_mac>
failover mac address inside <act_mac> <standby_mac>
failover mac address dmz <act_mac> <standby_mac>
```

Configuring the MAC addresses is not strictly necessary, however configuring them allows the standby PIX to acquire the MAC address of the active PIX in a failover situation. Since the MAC address will not change as part of failover, hosts communicating with the PIX during a failover will not need to wait for their ARP tables to timeout, creating more rapid recovery during a failure. Also, in the places that static ARP entries are used, the IP address used by the failover will never associate with an alternate MAC, which would effectively disable the ability to use failover if the PIX did not support acquisition of the active MAC address as well as the IP address. Note that there should not be a "failover mac" statement for the failover interface, as the failover interface does not require (or use) MAC acquisition.

State information for stateful failover must be passed over a LAN interface due to traffic volume. The next command defines the interface to use for stateful failover information:

```
failover link failover
```

While the interface specified for LAN failover technically can also be used as another segment interface, the volume of traffic generated by maintaining state between the PIX devices can be substantial. Therefore, a dedicated interface linked with a crossover cable is recommended. Also, the traffic may be substantial enough to require a gigabit Ethernet interface; however, this is not expected at the onset of operations for GIACE.

The next command shortens the failover polling interval from 15 to 5 to help speed up failover:

```
failover poll 5
```

Since we are required to pass state information over a LAN interface anyway, LAN based failover is configured. Cisco documentation recommends separate dedicated interfaces for LAN failover and state information respectively in high traffic installation; however, GIACE expects to see only average amounts of traffic after startup. If traffic becomes an issue on this interface, GIACE will either upgrade the failover interface to gigabit speeds or install another interface. The following command specifies the interface for LAN failover:

```
failover lan interface failover
```

Define this unit as the primary firewall:

```
failover lan unit primary
```

Keep stateful track of http traffic:

```
failover replicate http
```

This may provide little benefit since the failover time is several seconds for PIX firewalls. If this turns out to be the case, it may be preferable to increase performance on the firewall by disabling this feature.

The PIX does not currently replicate information for several protocols, the most notable of which to this implementation is ISAKMP and IPSec. This means that in a failover situation, VPN users would be required to reauthenticate, and the PIX would need to renegotiate keys with the logging and SNMP servers. This limitation has been accepted by GIACE.

Following the configuration of failover on the PIX, keeping the configurations in sync is critical. The following command, when issued on the active PIX, will synchronize the configuration between the active and standby devices appropriately:

```
write standby
```

When issued, the command will transfer an appropriate configuration from the running-config of the active PIX to the running-config of the standby PIX. Since the running-config is stored in RAM only, appropriate commands must be issued on each device to save the configuration to NVRAM to ensure that the configuration survives any reboots.

IPSec/VPN Configuration Commands

Due to the expected low utilization of remote VPN connections, the PIX firewalls will also be used to support the VPN. Two configurations are necessary: one for internal IPSec for the situations when the firewall must talk to an IPSec device such as for logging or SNMP, and the other for remote users connecting to the GIACE network from the Internet. First, we will discuss the configuration for the internal IPSec, then the configuration for the remote user IPSec. First we start by defining what mode IPSec is expected for the IPSec map defined:

```
crypto ipsec transform-set internal ah-md5-hmac mode transport
```

For internal clients, we will use transport mode Authentication Header (AH) for packet signing, since privacy of the traffic is less critical than integrity and AH will save some performance.

Enable IKE for key exchange:

```
crypto map internal 10 ipsec-isakmp
```

This command tells the PIX which address to use IPSec with:

```
crypto map internal 10 match address ipsec-internal
```

The Access Control List (ACL) “ipsec-internal” is defined in the Access List section next.

This command tells the PIX to use the “internal” transform set as defined above:

```
crypto map internal 10 set transform-set internal
```

Set the peer IP address for this map:

```
crypto map internal 10 set peer 192.168.193.0
```

The following command defines which interface to expect IPSec on. Since this map is strictly for traffic originating at the firewall and destined to the administrative segment, only the inside interface should be necessary:

```
crypto map internal 10 interface inside
```

The rest of this section deals with the configuration for remote VPN users. This is a little more complex configuration, since remote users are mobile and will not generally have a fixed IP address. Again, we start by defining in what mode IPSec is to perform for the remote user IPSec map:

```
crypto ipsec transform-set remote esp-aes esp-md5-hmac
```

Unlike the above transform set definition, the “remote” transform set forces AES in Encapsulating Security Payload (ESP) mode. This will encrypt all traffic for this map, ensuring both traffic integrity and traffic privacy as it traverses the hostile Internet. AES is both stronger and better performing than 3DES, the encryption algorithm with the current greatest install base.

Once again, activate IKE for key exchange:

```
crypto map remote 20 ipsec-isakmp dynamic dyn-remote
```

The remote user VPN configuration also includes configuration for the Authenex authentication server. Since most of the configuration for Authenex has been dealt with in the x section, only the assignment of the server to the VPN will be discussed here. This command configures this VPN map to authenticate to the radius server defined with the name “authenex”:

```
crypto map remote client token authentication authenex
```

The next command tells the PIX to initiate address negotiation with the remote client. This configuration will transfer IP configuration information to the remote client similar to DHCP on an open LAN:

```
crypto map remote client configuration address initiate
```

Define which interface on which the PIX should expect to see traffic for this IPSec map:

```
crypto map remote 20 interface outside
```

As above, this command defines which IP addresses are allowed to use the VPN. Since we are in tunnel mode with this configuration, this command also serves somewhat as access control:

```
crypto dynamic-map dyn-remote 20 match address ipsec-remote
```

Set the transform set for this map to be the group defined as “remote” above:

```
crypto dynamic-map dyn-remote 20 set transform-set remote
```

Enable Perfect Forward Secrecy (PFS). While PFS gives lesser performance, it does increase security by ensuring that new keys are generated for each negotiation:

```
crypto dynamic-map dyn-remote 20 set pfs group2
```

Enable ISAKMP:

```
isakmp enable outside
isakmp enable inside
```

Enable AES encryption for IKE as well:

```
isakmp policy 10 encryption aes
```

Set ISAKMP to use Diffie-Hellman (DH) group 2. Group 2 DH uses a 1024-bit key, making the key stronger than the 768-bit key used by group 1:

```
isakmp policy 10 group 2
```

Define the keys for remote and internal IPSec client, respectively:

```
isakmp key <strong_key1> address 0.0.0.0 netmask 0.0.0.0
isakmp key <strong_key2> address 192.168.193.0 netmask 255.255.255.0
```

Unique keys should definitely be assigned to each class of clients to limit exposure if a key should become compromised.

The following commands define the IP parameters assigned to remote VPN users when they connect:

```
vpngroup remote-ipsec address-pool remote-vpn
vpngroup remote-ipsec dns-server 192.168.129.20
vpngroup remote-ipsec dns-server 192.168.129.30
vpngroup remote-ipsec default-domain giace.com
vpngroup remote-ipsec split-tunnel ipsec-remote
vpngroup remote-ipsec idle-time 1800
vpngroup remote-ipsec max-time 43200
```

These commands are mostly self-explanatory, other than the first one. The first command defines the address pool to be assigned, and comes in the form of an access list defined in the Access List section below. The final two timeout commands are in seconds, which amounts to an idle time of 30 minutes and a maximum connect time of 12 hours.

Access List Commands

Cisco has designed the PIX OS to use Access Control Lists (ACLs) in a variety of situations, none the least of which is for controlling traffic crossing the PIX device. ACLs have replaced conduits in the PIX configuration for the preferred way to allow incoming traffic (for a PIX, traffic flowing from a lower security level interface to a higher level interface), and have added the flexibility to be able to control outgoing traffic (from a higher level interface to a lower level interface). ACLs may also be used for anywhere that a list of IP addresses might be necessary, such as with management access, VPN definitions, etc. In 6.2 and later PIX OS, a new feature called "Turbo ACLs" compiles the ACLs in order to speed up processing. Cisco claims that Turbo ACLs will not provide a performance increase until the ACL is at least 19 entries, however. Therefore, at the top of each ACL definition, there is one of the following commands:

```
access-list <name> compiled
no access-list <name> compiled
```

The first command activates Turbo-ACL; the second disables it. Note that all ACLs in the Cisco world default to deny-all, so even any packet not matching the list will automatically be denied (if the ACL is for traffic control) or the appropriate statement will not apply to the packet.

"ipsec-remote" ACL

The first ACL ("ipsec-remote") defines which IP addresses are allowed to use the remote IPSec VPN connections, and has been assigned to the "dyn-remote" IPSec definition in the IPSec/VPN section above:

```
no access-list ipsec-remote compiled
access-list ipsec-remote permit ip 192.168.0.0 255.255.0.0 192.168.253.0
      255.255.255.192
```

"ipsec-internal" ACL

The "ipsec-internal" ACL defines which IP addresses to use IPSec on the internal segment:

```
no access-list ipsec-internal compiled
access-list ipsec-internal permit ip 192.168.129.252 255.255.255.254
      192.168.193.0 255.255.255.0
```

"inbound" ACL

The "inbound" ACL is a traffic control ACL and gets a bit more complex. Each interface is allowed exactly one ACL assigned to filter traffic entering the interface, so the "inbound" ACL is applied to the "external" interface. Order does not matter in this list, since there is no overlap between statements.

This ACL is still not quite long enough for the use of Turbo-ACL:

```
no access-list inbound compiled
```

We need to allow HTTP and HTTPS to the informational website and E-Commerce websites, respectively:

```
access-list inbound permit tcp any host 1.2.3.10 eq www
access-list inbound permit tcp any host 1.2.3.11 eq 443
```

Allow incoming SMTP to the mail servers:

```
access-list inbound permit tcp any host 1.2.3.20 eq smtp
access-list inbound permit tcp any host 1.2.3.21 eq smtp
```

The Certificate Authority is a web application running over HTTPS:

```
access-list inbound permit tcp any host 1.2.3.50 eq 443
```

Allow IPSec AH from the border routers to the logging and SNMP hosts:

```
access-list inbound permit 51 1.2.3.252 255.255.255.254 host 1.2.3.241
access-list inbound permit 51 1.2.3.252 255.255.255.254 host 1.2.3.244
```

Allow IPSec AH from the IDS sensor to the IDS controller:

```
access-list inbound permit 51 host 1.2.3.240 host 1.2.3.242
```

Finally, apply the access list to the interface:

```
access-group inbound in interface external
```

"outbound-dmz" ACL

The "outbound-dmz" list consists of traffic controls for traffic leaving the DMZ, and is the first list which is big enough to optimize with Turbo-ACL:

```
access-list outbound-dmz compiled
```

First, block traffic that we want to never see traversing the firewall in any direction, but specifically want to log if the traffic should appear:

```
access-list outbound-dmz deny tcp any any eq 6667 log default
access-list outbound-dmz deny udp any any eq 67 log default
```

The list explicitly blocks all IRC and TFTP traffic due to their ready use as back channels, but others may be added in the future. Since this ACL is for the DMZ, the NetBIOS over TCP/IP (NBT) ports may also be candidates because of the high worm count, although logging NBT ports can be very noisy. The DMZ may be tolerable since there is only one Windows system; however, this should be tested and watched closely if implemented.

SMTP is tightly restricted to traffic only from two mail relays to the internal groupware server to limit the likelihood of becoming a SPAM relay or worm distributor:

```
access-list outbound-dmz permit tcp host 192.168.65.20 host 192.168.129.30 eq
      25
access-list outbound-dmz permit tcp host 192.168.65.21 host 192.168.129.30 eq
      25
```

Likewise, only the mail relays are allowed to talk directly to the Internet:

```
access-list outbound-dmz permit tcp host 192.168.65.20 any eq 25
access-list outbound-dmz permit tcp host 192.168.65.21 any eq 25
```

The E-Commerce servers are allowed to connect only to the fortune database:

```
access-list outbound-dmz permit tcp host 192.168.65.40 host 192.168.1.20 eq
      1433
access-list outbound-dmz permit tcp host 192.168.65.41 host 192.168.1.20 eq
      1433
```

Even though there are two database servers, the E-Commerce servers should only need to talk to the active IP in the cluster. Also, a reminder that all required information from the CRM database for E-Commerce operation is proxied by the fortune database cluster.

Next, allow the required IPSec ports for logging, SMTP, and IDS as described in the "inbound" list above:

```
access-list outbound-dmz permit 51 192.168.65.0 255.255.255.0 host
      192.168.193.11
access-list outbound-dmz permit 51 192.168.65.0 255.255.255.0 host
      192.168.193.14
access-list outbound-dmz permit 51 host 192.168.65.240 host 192.168.193.12
```

Now that all traffic allowed from the DMZ to the internal network (specifically the admin segment) is defined, block and log everything else to the internal network:

```
access-list outbound-dmz deny ip any 192.168.0.0 255.255.0.0 log
```

This statement makes the "outbound-dmz" ACL order-dependent. Any statement to allow traffic from the DMZ to the inside must be above this statement. Everything below this deny statement is a blanket to allow for traffic leaving the DMZ to the Internet:

```
access-list outbound-dmz permit udp host 192.168.65.20 any eq 53
access-list outbound-dmz permit udp host 192.168.65.21 any eq 53
access-list outbound-dmz permit udp host 192.168.65.20 any eq 123
access-list outbound-dmz permit udp host 192.168.65.21 any eq 123
access-list outbound-dmz permit tcp host 192.168.65.20 any eq 80
access-list outbound-dmz permit tcp host 192.168.65.21 any eq 80
access-list outbound-dmz permit tcp host 192.168.65.20 any eq 443
access-list outbound-dmz permit tcp host 192.168.65.21 any eq 443
```

Obviously, DNS and NTP are necessary to provide name resolution and time synchronization from the Internet at large. Similarly, HTTP and HTTPS from the proxy servers are required, since the proxy servers provide a gateway for all of these services to the rest of the GIACE network.

The last step is to apply the access list to the DMZ interface:

```
access-group outbound-dmz in interface dmz
```

"outbound-dmz ACL"

The "outbound-dmz" ACL is the lengthiest of the PIX ACLs. Again, we start by enabling Turbo-ACL and filtering out traffic we never want to see, but want logged:

```
access-list outbound-internal compiled
access-list outbound-internal deny tcp any any eq 6667 log default
access-list outbound-internal deny udp any any eq 67 log default
```

Next, allow the groupware and file servers to access the Internet for DNS and NTP:

```
access-list outbound-internal permit udp host 192.168.129.20 any eq 53
access-list outbound-internal permit udp host 192.168.129.30 any eq 53
access-list outbound-internal permit udp host 192.168.129.20 any eq 123
access-list outbound-internal permit udp host 192.168.129.30 any eq 123
```

Now, we allow the logging and SNMP servers to send email directly:

```
access-list outbound-internal permit tcp host 192.168.193.11 any eq 25
access-list outbound-internal permit tcp host 192.168.193.12 any eq 25
```

These servers are allowed to send directly to allow for paging and alerting in case the firewall is down, the mail relays are down, or the proxy servers are under attack.

Allow all internal nodes access to the proxy servers:

```
access-list outbound-internal permit tcp 192.168.0.0 255.255.0.0 host
      192.168.65.20 eq 3128
```

```
access-list outbound-internal permit tcp 192.168.0.0 255.255.0.0 host
      192.168.65.21 eq 3128
```

Allow IPSec traffic to the DMZ and appropriate external hosts:

```
access-list outbound-internal permit 51 host 192.168.193.11 192.168.65.0
      255.255.255.0
access-list outbound-internal permit 51 host 192.168.193.14 192.168.65.0
      255.255.255.0
access-list outbound-internal permit 51 host 192.168.193.11 1.2.3.252
      255.255.255.254
access-list outbound-internal permit 51 host 192.168.193.14 1.2.3.252
      255.255.255.254
access-list outbound-internal permit 51 host 192.168.193.12 host 1.2.3.240
```

Allow direct connections to DMZ hosts via SSH, HTTP and HTTPS for administration and troubleshooting:

```
access-list outbound-internal permit tcp any 192.168.65.0 255.255.255.0 eq 22
access-list outbound-internal permit tcp any 192.168.65.0 255.255.255.0 eq 22
access-list outbound-internal permit tcp any 192.168.65.0 255.255.255.0 eq 80
access-list outbound-internal permit tcp any 192.168.65.0 255.255.255.0 eq 80
access-list outbound-internal permit tcp any 192.168.65.0 255.255.255.0 eq
      443
access-list outbound-internal permit tcp any 192.168.65.0 255.255.255.0 eq
      443
```

Finally, apply the ACL to the internal interface:

```
access-group outbound-internal in interface internal
```

Internal Routers

The internal routers are also Cisco devices running Cisco IOS version 12.3. Therefore, many of the commands are similar to those outlined in the border router section above. Specifically, the "service" Commands, Logging & SNMP Configuration Commands, and Device Access Configuration Commands groups of commands are almost identical, and therefore not repeated here. The IP & Routing Configuration section does not contain the full set of null routes in order to keep the routing table small. The border router and firewalls are both configured to block this sort of traffic, so a third time is unnecessary and will potentially slow up the highly performance sensitive core routers. Otherwise, the IP & Routing Configuration section is identical.

The interface configurations are slightly different however, due to the Catalyst 6500/MSFC hardware platform. The ACL sets for the internal routers also take a completely different approach than those of the border routers. These sections of the configuration will be covered in detail here. A complete configuration for an internal router MSFC cards is included in Appendix D.

Interface Configuration Commands

Like with the border routers above, we will use the null interface to take care of some undesirable packets. Therefore we will initialize it and squash ICMP unreachable messages, just as above:

```
interface Null 0
 no ip unreachables
```

The biggest difference in interface configuration from the border routers is that all interfaces on the MSFC are virtual interfaces. All interfaces will connect to a VLAN, but VLANs must be already created on the switching side of the Catalyst 6500 prior to configuration on the MSFC. Configuration of the 6500 is beyond the scope of this document, so the reader may assume that VLANs 1, 65, 129, and 193 are already created. The VLAN numbers chosen match the 3rd octet of the IP range assigned to the network for administrative convenience. Other than the fact that all interfaces are VLAN interfaces, the configuration is fairly similar:

```
interface vlan193
 description Administrative VLAN Interface
 ip address 192.168.193.253 255.255.255.0
 ip access-group antinoise in
 ip access-group to-admin out
 no ip proxy-arp
 ip accounting access-violations
 ip route-cache flow
 no cdp enable
 standby authentication admin
 standby name GIACE-admin-gw
 standby ip 192.168.193.1
```

Since this is an internal network, some of the interface settings are more relaxed for administrative convenience over security. Most notably:

- ICMP Unreachable messages are not suppressed.
- ICMP Redirect messages are also allowed on this network in order to allow more efficient processing Internet-bound traffic to the firewall.
- Multicast is not suppressed within the confines of the internal network.

Access Control Lists (ACLs)

While the interface configurations vary slightly from the border routers, the ACLs take a drastically different approach. While the border routers are configured to filter a little traffic and pass the rest through (known as default accept), the internal routers are configured to pass a little and filter anything else (default block). The border routers are configured to block before routing, but the multiple interfaces on the internal routers dictate a route, and then block approach to function as desired.

Anti-Noise

The first ACL is designed to filter some traffic that is undesirable on the internal network, and is to be applied to all interfaces. This is a subset of the "antinoise" filter used on the border routers:

```
no ip access-list extended antinoise
ip access-list extended antinoise
 remark Knocks down traffic that we never want to cross the core router
 remark Apply to in on all interfaces
 remark --- no use for tcp/udp small services
 deny   tcp any any range 1 19 log-input
 deny   udp any any range 1 19 log-input
 remark --- TFTP has some uses - may want to enable on a case-by-case
 deny   udp any any eq 69 log-input
 remark --- IRC Bad - control channel for worms
 deny   tcp any any eq 194 log-input
 deny udp any any eq 194 log-input
 deny tcp any any eq 994 log-input
 deny udp any any eq 994 log-input
 deny tcp any any eq 6667 log-input
 deny udp any any eq 6667 log-input
 deny tcp any any eq 7776 log-input
 remark --- let the rest at least get to the outbound ACL
 permit ip any any
```

In summary, this ACL blocks small services, and IRC since they provide no useful purpose on the GIACE network. TFTP is also blocked, and although it may be enabled from time to time for network device management, TFTP usage across subnets should be extremely rare. Since this ACL is applied inbound to each interface, it will have the effect of limiting traffic leaving the VLAN. (This seems backwards, but in- and out-bound are defined from the perspective of the interface.) Therefore, we want to pass all traffic that might possibly need to pass to another network, hence the "permit ip any any" at the end of the ACL.

Database VLAN

The database VLAN should only see one user traffic flow: database connections. Administratively, we need a few more traffic flows for time synchronization, name services, and connection to the Active Directory for authentication. The ACL looks as follows:

```
no ip access-list extended to-db
ip access-list extended to-db
 remark This access list defines traffic allowed to enter the DB vlan
 remark Apply to VLAN1 out
 permit tcp any any established
 remark --- Allow database access from all internal hosts
 permit tcp 192.168.0.0 0.0.255.255 any eq 1433
 permit udp 192.168.0.0 0.0.255.255 any eq 1434
 remark --- Allow NTP time sync
 permit udp host 192.168.129.21 any eq 123
 permit udp host 192.168.129.31 any eq 123
```

```
 remark --- Allow DNS query responses
 permit udp host 192.168.129.21 any eq 53
 permit udp host 192.168.129.31 any eq 53
 remark --- Allow a host of ports for AD
 permit udp host 192.168.129.21 any eq 135
 permit udp host 192.168.129.21 any eq 88
 permit udp host 192.168.129.21 any eq 389
 remark --- Backup software should connect over above defined SQL ports
 remark --- If more access necessary, add here
 deny   ip any any log-input
```

The ACL is pretty straightforward, with the exception of two commands. The "permit tcp any any established" command allows traffic from TCP connections already established to return, effectively allowing free outbound connections from this VLAN. The other is "deny ip any any log-input." While Cisco ACLs automatically have a default action of deny, this does not log anything for the dropped packets. Adding this command allows the router to log all dropped packets. This ACL will be applied to the database VLAN interface (VLAN1) in the outbound connection so that it only filters traffic arriving on the database VLAN.

Administrative VLAN

The administrative VLAN has more complex traffic flow requirements than the database VLAN; however, the principles are more or less the same:

```
no ip access-list extended to-admin
ip access-list extended to-admin
 remark This access list defined traffic allowed to enter the ADMIN vlan
 remark Apply to VLAN193 out - not yet performance tuned!
 remark All hosts on the ADMIN network are implied permit, including the
      MSFCs
 remark This list only contains currently defined devices - there will be
      more!
 permit tcp any any established
 remark --- Allow syslog only from particular devices - don't want the
      possiblity of a DoS!
 permit udp 1.2.3.252 0.0.0.1 host 192.168.193.14 eq 514
 permit udp 192.168.1.20 0.0.0.1 host 192.168.193.14 eq 514
 permit udp 192.168.1.40 0.0.0.1 host 192.168.193.14 eq 514
 permit udp 192.168.65.20 0.0.0.1 host 192.168.193.14 eq 514
 permit udp 192.168.65.30 0.0.0.1 host 192.168.193.14 eq 514
 permit udp 192.168.65.40 0.0.0.1 host 192.168.193.14 eq 514
 permit udp host 192.168.65.50 host 192.168.193.14 eq 514
 permit udp host 192.168.65.60 host 192.168.193.14 eq 514
 permit udp 192.168.65.250 0.0.0.1 host 192.168.193.14 eq 514
 permit udp 192.168.65.252 0.0.0.1 host 192.168.193.14 eq 514
 permit udp host 192.168.129.10 host 192.168.193.14 eq 514
 permit udp host 192.168.129.20 host 192.168.193.14 eq 514
 permit udp host 192.168.129.30 host 192.168.193.14 eq 514
 permit udp 192.168.129.252 0.0.0.1 host 192.168.193.14 eq 514
 remark --- Allow time/DNS responses from appropriate servers
 permit udp host 192.168.129.20 any eq 53
 permit udp host 192.168.129.30 any eq 53
```

```
permit udp host 192.168.129.20 any eq 123
permit udp host 192.168.129.30 any eq 123
remark --- Allow a host of ports for AD
permit udp host 192.168.129.21 any eq 135
permit udp host 192.168.129.21 any eq 88
permit udp host 192.168.129.21 any eq 389
remark --- Allow SNMP information to monitor station
permit udp 1.2.3.252 0.0.0.1 host 192.168.193.11 range 161 162
permit udp 192.168.1.20 0.0.0.1 host 192.168.193.11 range 161 162
permit udp 192.168.1.40 0.0.0.1 host 192.168.193.11 range 161 162
permit udp 192.168.65.20 0.0.0.1 host 192.168.193.11 range 161 162
permit udp 192.168.65.30 0.0.0.1 host 192.168.193.11 range 161 162
permit udp 192.168.65.40 0.0.0.1 host 192.168.193.11 range 161 162
permit udp host 192.168.65.50 host 192.168.193.11 range 161 162
permit udp host 192.168.65.60 host 192.168.193.11 range 161 162
permit udp 192.168.65.250 0.0.0.1 host 192.168.193.11 range 161 162
permit udp 192.168.65.252 0.0.0.1 host 192.168.193.11 range 161 162
permit udp host 192.168.129.10 host 192.168.193.11 range 161 162
permit udp host 192.168.129.20 host 192.168.193.11 range 161 162
permit udp host 192.168.129.30 host 192.168.193.11 range 161 162
permit udp 192.168.129.252 0.0.0.1 host 192.168.193.11 range 161 162
remark --- Allow RADIUS from firewalls to auth server
permit tcp 192.168.129.252 0.0.0.1 host 192.168.193.15 range 1812 1813
permit udp 192.168.129.252 0.0.0.1 host 192.168.193.15 range 1812 1813
remark --- Allow IDS sensor logging to IDS console - runs on mysql
permit tcp host 1.2.3.240 host 192.168.193.12 eq 3306
permit tcp host 192.168.65.240 host 192.168.193.12 eq 3306
remark --- backups should be all outbound connections, if not add rules here
deny   ip any any log-input
```

This ACL allows traffic flows for logging, SNMP management, VPN authentication from the firewall, and IDS logging, as well as traffic for administrative functions such as time synchronization, name resolution, and Active Directory.

Linux Host Firewalls

The following rule sets are for the host firewall configurations for the DMZ servers. Again, one configuration is provided for each host in an otherwise identical cluster. All of the Linux machines presented here will have routing disabled, therefore the INPUT and OUTPUT tables perform exactly as named, and the FORWARD table is unused. A sample of a monolithic load script and the results is included in Appendix E. GIACE will probably wish to use a hierarchical script approach where each table is contained within its own file in order to facilitate easier management of changes to the common tables.

Common Tables

This section treats the tables that all hosts have in common. Custom tables are used throughout the Linux configurations to aide in readability and performance. In this document, these tables are linked to improve the readability the configuration.

BAN

This table contains lists of valid IP addresses that have been banned for some reason. Right now it contains a noisy ISP, but will be modified in the course of operation of the GIACE network to add blocks that perform undesirable actions against GIACE (such as chronic port scanners, etc.). Care must be taken to ensure that banned sites do not have a business need to communicate with GIACE!

```
# Create the BAN table
iptables -N BAN

# A couple noisy ISP blocks - SAVECOM-NET, UNICOM
iptables -A BAN -s 61.65.0.0/255.255.128.0 -j LOGDROP
iptables -A BAN -s 61.240.0.0/255.252.0.0 -j LOGDROP
# Return if no match
```

INVAL_SRC

The purpose of this table is to collect all the illegal and unused IP blocks that should be filtered to prevent spoofing.[1] Each block gets two rules: the first is to log the occurrence, the second to drop it. Blocks are aggregated into the smallest CIDR block possible. This makes the list less human-readable and harder to maintain; however, since each of these rules will be passed for virtually every packet, it will provide a significant performance increase. Since we know are behind the NAT device, we know that 192.168.[1,65,128-129,193].x are valid. We do not want to accept anything directly here, since there is more processing that we wish to do, hence the “RETURN” instead of “ACCEPT.” This table should remain relatively static, changing only periodically when IP address assignments change.

```
# Create the INVAL_SRC table
iptables -N INVAL_SRC

# Short circuit known good addresses for better performance.
# We may elect to filter more of these with later rules.
# These rules are order independent among each other,
# but all should remain at the top for performance.
iptables -A INVAL_SRC -s 192.168.1.0/255.255.255.0 -j RETURN
iptables -A INVAL_SRC -s 192.168.65.0/255.255.255.0 -j RETURN
iptables -A INVAL_SRC -s 192.168.129.0/255.255.254.0 -j RETURN
iptables -A INVAL_SRC -s 192.168.193.0/255.255.255.0 -j RETURN

# Start stripping illegal and unassigned addresses.
# This is not the place to block naughty but valid addresses -
# the BAN table exists for that.
# The following rules may be reordered at will if necessary.
iptables -A INVAL_SRC -s 1.0.0.0/255.0.0.0 -j LOGDROP
iptables -A INVAL_SRC -s 2.0.0.0/255.0.0.0 -j LOGDROP
iptables -A INVAL_SRC -s 5.0.0.0/255.0.0.0 -j LOGDROP
```

[1] IP Blocks for this chain were taken from “Secure IOS Template v3.1” by Rob Thomas, modified for IPTABLES and aggregated. This template is regularly maintained by Thomas and is available at http://www.cymru.com/Documents/secure-ios-template.html.

```
iptables -A INVAL_SRC -s 7.0.0.0/255.0.0.0 -j LOGDROP
iptables -A INVAL_SRC -s 10.0.0.0/255.0.0.0 -j LOGDROP
iptables -A INVAL_SRC -s 23.0.0.0/255.0.0.0 -j LOGDROP
iptables -A INVAL_SRC -s 27.0.0.0/255.0.0.0 -j LOGDROP
iptables -A INVAL_SRC -s 31.0.0.0/255.0.0.0 -j LOGDROP
iptables -A INVAL_SRC -s 36.0.0.0/254.0.0.0 -j LOGDROP
iptables -A INVAL_SRC -s 39.0.0.0/255.0.0.0 -j LOGDROP
iptables -A INVAL_SRC -s 41.0.0.0/255.0.0.0 -j LOGDROP
iptables -A INVAL_SRC -s 42.0.0.0/255.0.0.0 -j LOGDROP
iptables -A INVAL_SRC -s 49.0.0.0/255.0.0.0 -j LOGDROP
iptables -A INVAL_SRC -s 50.0.0.0/255.0.0.0 -j LOGDROP
iptables -A INVAL_SRC -s 58.0.0.0/254.0.0.0 -j LOGDROP
iptables -A INVAL_SRC -s 70.0.0.0/254.0.0.0 -j LOGDROP
iptables -A INVAL_SRC -s 72.0.0.0/248.0.0.0 -j LOGDROP
iptables -A INVAL_SRC -s 83.0.0.0/255.0.0.0 -j LOGDROP
iptables -A INVAL_SRC -s 84.0.0.0/252.0.0.0 -j LOGDROP
iptables -A INVAL_SRC -s 88.0.0.0/248.0.0.0 -j LOGDROP
iptables -A INVAL_SRC -s 96.0.0.0/224.0.0.0 -j LOGDROP
iptables -A INVAL_SRC -s 169.254.0.0/255.255.0.0 -j LOGDROP
iptables -A INVAL_SRC -s 172.16.0.0/255.240.0.0 -j LOGDROP
iptables -A INVAL_SRC -s 173.0.0.0/255.0.0.0 -j LOGDROP
iptables -A INVAL_SRC -s 174.0.0.0/254.0.0.0 -j LOGDROP
iptables -A INVAL_SRC -s 176.0.0.0/248.0.0.0 -j LOGDROP
iptables -A INVAL_SRC -s 184.0.0.0/252.0.0.0 -j LOGDROP
iptables -A INVAL_SRC -s 189.0.0.0/255.0.0.0 -j LOGDROP
iptables -A INVAL_SRC -s 190.0.0.0/255.0.0.0 -j LOGDROP
iptables -A INVAL_SRC -s 192.0.2.0/255.255.255.0 -j LOGDROP
iptables -A INVAL_SRC -s 192.168.0.0/255.255.0.0 -j LOGDROP
iptables -A INVAL_SRC -s 197.0.0.0/255.0.0.0 -j LOGDROP
iptables -A INVAL_SRC -s 223.0.0.0/255.0.0.0 -j LOGDROP
# Return to calling rule for more processing.
```

LOGDROP

This list simply logs a packet and then drops it. Jumping to a new table rather than including two rule entries in the parent chain means that only one rule is evaluated for each packet that doesn't match. Three rules must be evaluated instead of two for packets that match, however. Since something is very wrong if a DMZ host ever sees this type of traffic, rule matches to jump here should be rare; therefore, formatting the rule set in this manner should provide the best performance. This rule ends with a jump to DROP, which is terminal, so this rule should never return. "HOSTNAME" will be replaced with the hostname of the individual machine in order to more easily identify which machines are logging. More specific prefixes would be nice, however are not available here because this table is used from several locations, and IPTABLES does not support passing of variables between tables. Variables are acceptable in the shell script used to load the tables however, hence the reference to the $LOC_HOSTNAME variable, which must be set at the start of the script.

```
# Create the LOGDROP table
iptables -N LOGDROP

# Log every packet that reaches this table verbosely,
# prefixing with the system hostname.
```

```
iptables -A LOGDROP -j LOG --log-prefix $LOC_HOSTNAME --log-tcp-options --
      log-ip-options

# Drop every packet that reaches this table.
iptables -A LOGDROP -j DROP
```

INPUT

The INPUT chain for all servers is the same for all hosts, and assumes that the appropriate user-defined tables LOGDROP, BAN, INVAL_SRC, and VALID are already created and loaded (in that order):

```
# Main INPUT table
# Look for banned sites in the BAN table first.  There is a reason they are
      banned!
iptables -A INPUT -j BAN

# Next strip RFC-1918, other invalid, or unassigned IP source ranges as
      defined in the INVAL_SRC table.
iptables -A INPUT -j INVAL_SRC

# Trash packets for connections not properly registered in the state table.
      (I.E. SYN-FIN, etc.)
iptables -A INPUT -m state --state INVALID -j LOGDROP

# Disallow fragments - the routers should have reassembled already
iptables -A INPUT -f -j LOGDROP

# Look for allowable traffic on the VALID table
iptables -A INPUT -j VALID

# Log and drop everything else
iptables -A INPUT -j LOGDROP
```

These rules should, for the most part, stay in order. The first four lines may be rearranged slightly if performance dictates, but generally they shouldn't be triggered much. The last two must remain exactly as is or the rule set behavior will change dramatically!

Informational Web Server

The VALID table is the only table specific to each machine and contains the list of traffic allowed to the machine. It assumes that $LOC_IP is set with the IP of the local machine.

```
# VALID table contains the only rules that should be changed from machine
# to machine.  All others tables are standard within the DMZ machines.
# VALID is also the only table which should contain accept rules.
# Create the VALID table
iptables -N VALID

# Heartbeat for HA clustering.  HA runs multicast, so this rule can be
```

```
# further limited to a single address for each cluster once HA is configured.
iptables -A VALID -p udp -m udp -d $LOC_IP --dport 694 -j ACCEPT

# Obviously need HTTP allowed
iptables -A VALID -p tcp -m tcp -d $LOC_IP --dport 80 -j ACCEPT

# Time/DNS are UDP and therefore not stateful, so we must
# explicitly accept responses:
iptables -A VALID -p udp -m udp -s 192.168.65.20 -d $LOC_IP --dport 123 -j
      ACCEPT
iptables -A VALID -p udp -m udp -s 192.168.65.21 -d $LOC_IP --dport 123 -j
      ACCEPT
iptables -A VALID -p udp -m udp -s 192.168.65.20 -d $LOC_IP --dport 53 -j
      ACCEPT
iptables -A VALID -p udp -m udp -s 192.168.65.21 -d $LOC_IP --dport 53 -j
      ACCEPT

# IPSec for logging and other admin stuff
iptables -A VALID -p 50 -s 192.168.193.0/24 -d $LOC_IP -j ACCEPT
iptables -A VALID -p 51 -s 192.168.193.0/24 -d $LOC_IP -j ACCEPT
iptables -A VALID -p udp -m udp -s 192.168.193.0/24 --sport 500 -d $LOC_IP --
      dport 500 -j ACCEPT

# SSH for management - only from monitor workstation.
# Might add select administrator workstations later.
iptables -A VALID -p tcp -m tcp -s 192.168.193.11 -d $LOC_IP --dport 22 -j
      ACCEPT
```

These rules may be moved around within the table, however they are roughly ordered for performance at this time. Further performance tuning may be necessary later.

E-Commerce Web Server

The VALID table for the E-Commerce servers is similar:

```
# VALID table contains the only rules that should be changed from machine
# to machine.  All others tables are standard within the DMZ machines.
# VALID is also the only table which should contain accept rules.
# Create the VALID table
iptables -N VALID

# Heartbeat for HA clustering.  HA runs multicast, so this rule can be
# further limited to a single address for each cluster once HA is configured.
iptables -A VALID -p udp -m udp --dport 694 -j ACCEPT

# Obviously need HTTP(S) allowed
iptables -A VALID -p tcp -m tcp --dport 80 -j ACCEPT
iptables -A VALID -p tcp -m tcp --dport 443 -j ACCEPT

# Time/DNS are UDP and therefore not stateful, so we must
# explicitly allow to receive responses:
iptables -A VALID -p udp -m udp -s 192.168.65.20 --dport 123 -j ACCEPT
iptables -A VALID -p udp -m udp -s 192.168.65.21 --dport 123 -j ACCEPT
iptables -A VALID -p udp -m udp -s 192.168.65.20 --dport 53 -j ACCEPT
iptables -A VALID -p udp -m udp -s 192.168.65.21 --dport 53 -j ACCEPT
```

```
# IPSec for logging and other admin stuff
iptables -A VALID -p 50 -s 192.168.193.0/255.255.255.0 -j ACCEPT
iptables -A VALID -p 51 -s 192.168.193.0/255.255.255.0 -j ACCEPT
iptables -A VALID -p udp -m udp -s 192.168.193.0/255.255.255.0 --sport 500 --
      dport 500 -j ACCEPT

# SSH for management - only from monitor workstation.
# Might add select administrator workstations later.
iptables -A VALID -p tcp -m tcp -s 192.168.193.11 --dport 22 -j ACCEPT
```

These rules also may be moved around within the table, however they are roughly ordered for performance at this time. Further performance tuning may be necessary later.

Proxy Server/SMTP Gateway

The VALID table for the Proxy servers varies a bit more:

```
# VALID table contains the only rules that should be changed from machine
# to machine.  All others tables are standard within the DMZ machines.
# VALID is also the only table which should contain accept rules.
# Create the VALID table
iptables -N VALID

# Heartbeat for HA clustering.  HA runs multicast, so this rule can be
# further limited to a single address for each cluster once HA is configured.
iptables -A VALID -p udp -m udp --dport 694 -j ACCEPT

# Time services are allowed to upstream time servers and everywhere local
# Time services may be adjusted to more locally appropriate servers once ISP
# is set
iptables -A VALID -p udp -m udp -s 199.240.130.1 --dport 123 -j ACCEPT
iptables -A VALID -p udp -m udp -s 216.204.130.12 --dport 123 -j ACCEPT
iptables -A VALID -p udp -m udp -s 192.168.65.0/255.255.255.0 --dport 123 -j
      ACCEPT

# DNS allowed to upstream servers (BellSouth for now) and everywhere local
# DNS services should be adjusted to values provided by ISP when ISP is set.
iptables -A VALID -p udp -m udp -s 205.152.0.8 --dport 53 -j ACCEPT
iptables -A VALID -p udp -m udp -s 205.152.144.188 --dport 53 -j ACCEPT
iptables -A VALID -p udp -m udp -s 192.168.65.0/255.255.255.0 --dport 53 -j
      ACCEPT

# IPSec for logging and other admin stuff
iptables -A VALID -p 50 -s 192.168.193.0/255.255.255.0 -j ACCEPT
iptables -A VALID -p 51 -s 192.168.193.0/255.255.255.0 -j ACCEPT
iptables -A VALID -p udp -m udp -s 192.168.193.0/255.255.255.0 --sport 500 --
      dport 500 -j ACCEPT

# Obviously need SMTP allowed
iptables -A VALID -p tcp -m tcp --dport 25 -j ACCEPT

# Proxy stuff - "RELATED" rule is because the proxy may require
# helper ports (such as TCP-DATA) to connect back.
```

```
iptables -A VALID -p tcp -m tcp -s 192.168.0.0/255.255.0.0 --dport 3128 -j
      ACCEPT
iptables -A VALID -m state -state RELATED -j ACCEPT

# SSH for management - only from monitor workstation.
# Might add select administrator workstations later.
iptables -A VALID -p tcp -m tcp -s 192.168.193.11 --dport 22 -j ACCEPT
```

Once again, these rules may be moved around within the table; however, they are roughly ordered for performance at this time. Further performance tuning may be necessary later.

DMZ Switch

The Cisco Catalyst 3550 switch is an IOS based switch. Therefore, many of the configuration considerations for the Cisco Router configurations also apply to the 3550. Rather than repeating them here, they may be reviewed in the Border Routers section above. Relevant sections are: "service" Commands, Logging & SNMP Configuration Commands, and Device Access Configuration Commands.

Since the device is a switch instead of a router, the interface configuration is somewhat different, however many of the commands are exactly the same. In order for the switch to be able to communicate in-band with IP, a VLAN interface must be defined. At first glance, since all management of the switch is out-of-band, it may appear that this step is unnecessary. Without an IP address, the switch will not be able to send SNMP or logging information, however, necessitating the need for this configuration.

The VLAN interface is defined with the following commands functioning exactly as before:

```
interface VLAN1
 description Internal Interface
 ip address 192.168.65.250
 no ip route-cache
 no ip unreachables
```

One commands are added however in order to help lock in communication on the switch. The following command prevents ARP table entries from timing out:

```
arp timeout 0
```

This makes it nearly impossible to perform ARP poisoning attacks against the switch, as the switch will never forget ARP entries. (The switch should only ever talk to the PIX firewalls for SNMP and logging purposes). A permanent ARP cache also gives the benefit of recording all hosts on the local segment that have ever attempted to talk to the switch, such as by a ping sweep, port scan, or telnet attempt. This has very limited benefits if not checked regularly, but may be useful if questions arise about the behavior of machines in the DMZ.

On IOS based switches such as the 3550, each port is defined as an interface as well. While Cisco did this to provide a "unified" interface for all Cisco platforms, it forces administrators to administer each port separately, rather than providing the ability to change settings by range as is available on the Catalyst OS based switches. It does, however, make for a familiar syntax, and the following commands perform exactly the same as on a Cisco router:

```
interface FastEthernet0/1
 description Firewall_Default Gateway
 no cdp enable
```

In order to prevent ARP spoofing activities, we want to enable port security. The following commands limit the port to accept only frames from the first learned MAC address, and to send warning traps if frames from additional MAC addresses arrive via this port.

```
port security max-mac-count 1
port security action trap
```

Cisco also has the option to automatically shutdown the port when a MAC violation is detected. Shutting down the port will create two possibly undesirable effects however:

- Performing a port shutdown will deny all service to the server until the port is administratively re-enabled. This is not desirable in a high-availability environment.
- Alerting only provides greater opportunity to investigate suspicious behavior without alerting an attacker if the violation is caused by an active attack. ARP -based attacks are very suspicious and should be taken seriously!

This command ensures that VLAN trunking on all ports is disabled. Since these switches are not configured with VLANs to begin with, this should be moot; however it will be included for thoroughness.

```
switchport mode access
```

The final command in an active port configuration allows the machine to bypass Spanning Tree states and go straight to forwarding, eliminating a 50 second delay before the machine would otherwise be allowed to pass traffic. This is safe only because a single machine is connected to each port. The following command should not be issued on ports connected to other switches or hubs:

```
spanning-tree portfast
```

The above commands will be issued on all ports on each switch, whether the port is currently in use or vacant. Pre-configuration ensures that administrators do not forget to issue the appropriate configuration commands when activating a new port. To protect against unauthorized physical connection to vacant ports, each vacant port will

also be disabled with the “shutdown” command. To activate a disabled port, simply use the following:

```
no shutdown
```

When combined, configuration of an active port will look similar to the following:

```
interface FastEthernet0/1
 description Firewall_Default Gateway
 port security max-mac-count 1
 port security action trap
 spanning-tree portfast
 no cdp enable
```

A vacant port will look like this:

```
interface FastEthernet0/10
 shutdown
 port security max-mac-count 1
 port security action trap
 spanning-tree portfast
 no cdp enable
```

A complete configuration for a DMZ switches is included in Appendix F.

Implementation Tutorial – PIX

Implementation of PIX configurations is usually as simple as loading the text-based configuration file, either through TFTP, cut and paste, or by hand. In this case, the redundant configuration adds a few caveats however. While the PIX allows for synchronizing configurations between two firewalls in a failover configuration, not all commands in the configuration are synchronized. To avoid determining which commands are synchronized and which are not, this tutorial will generally issue the commands independently on both the active and standby firewalls, rather than expecting the firewall to synchronize all commands.

This tutorial assumes that the PIXs in question are new firewalls with exactly 4 Ethernet interfaces, and properly licensed for redundancy and AES IPSec. Also, the purpose of this tutorial is not to explain the purpose of each command, but to provide instructions for configuration of the PIX firewalls. For interpretation of each individual command, please refer to the Firewalls section or the Cisco PIX Command Reference.

First, each firewall must be configured with the basic configuration information unique to the firewall, such as IP addresses, and hostnames. Issuing the following commands will perform the required configuration on the active firewall:

```
interface ethernet0 100full
interface ethernet1 100full
interface ethernet2 100full
```

```
interface ethernet3 100full
nameif ethernet0 outside security0
nameif ethernet1 dmz security20
nameif ethernet2 inside security100
nameif ethernet3 failover security10
enable password <GIACEpw>
password <GIACEpw>
hostname GIACE-fw-a
domain-name giace.com
clock timezone EST -5
fixup protocol dns maximum-length 512
fixup protocol ftp
fixup protocol http 80
no fixup protocol h323 h225
no fixup protocol h323 ras
no fixup protocol ils
no fixup protocol rsh
fixup protocol rtsp 554
fixup protocol smtp 25
no fixup protocol sqlnet 1521
no fixup protocol sip 5060
no fixup protocol skinny 2000
no access-list inbound compiled
access-list inbound permit tcp any host 1.2.3.10 eq www
access-list inbound permit tcp any host 1.2.3.11 eq 443
access-list inbound permit tcp any host 1.2.3.20 eq smtp
access-list inbound permit tcp any host 1.2.3.21 eq smtp
access-list inbound permit tcp any host 1.2.3.50 eq 443
access-list inbound permit 51 1.2.3.252 255.255.255.254 host 1.2.3.241
access-list inbound permit 51 1.2.3.252 255.255.255.254 host 1.2.3.244
access-list inbound permit 51 host 1.2.3.240 host 1.2.3.242
no access-list ipsec-remote compiled
access-list ipsec-remote permit ip 192.168.0.0 255.255.0.0 192.168.253.0
      255.255.255.192
no access-list ipsec-internal compiled
access-list ipsec-internal permit ip 192.168.129.252 255.255.255.254
      192.168.193.0 255.255.255.0
access-list outbound-dmz compiled
access-list outbound-dmz deny tcp any any eq 6667 log default
access-list outbound-dmz deny udp any any eq 67 log default
access-list outbound-dmz permit tcp host 192.168.65.20 host 192.168.129.30 eq
      25
access-list outbound-dmz permit tcp host 192.168.65.21 host 192.168.129.30 eq
      25
access-list outbound-dmz permit tcp host 192.168.65.40 host 192.168.1.20 eq
      1433
access-list outbound-dmz permit tcp host 192.168.65.41 host 192.168.1.20 eq
      1433
access-list outbound-dmz permit 51 192.168.65.0 255.255.255.0 host
      192.168.193.11
access-list outbound-dmz permit 51 192.168.65.0 255.255.255.0 host
      192.168.193.14
access-list outbound-dmz permit 51 host 192.168.65.240 host 192.168.193.12
access-list outbound-dmz deny ip any 192.168.0.0 255.255.0.0 log
access-list outbound-dmz permit udp host 192.168.65.20 any eq 53
access-list outbound-dmz permit udp host 192.168.65.21 any eq 53
access-list outbound-dmz permit udp host 192.168.65.20 any eq 123
```

```
access-list outbound-dmz permit udp host 192.168.65.21 any eq 123
access-list outbound-dmz permit tcp host 192.168.65.20 any eq 80
access-list outbound-dmz permit tcp host 192.168.65.21 any eq 80
access-list outbound-dmz permit tcp host 192.168.65.20 any eq 443
access-list outbound-dmz permit tcp host 192.168.65.21 any eq 443
access-list outbound-dmz permit tcp host 192.168.65.20 any eq 25
access-list outbound-dmz permit tcp host 192.168.65.21 any eq 25
access-list outbound-internal compiled
access-list outbound-internal deny tcp any any eq 6667 log default
access-list outbound-internal deny udp any any eq 67 log default
access-list outbound-internal permit udp host 192.168.129.20 any eq 53
access-list outbound-internal permit udp host 192.168.129.30 any eq 53
access-list outbound-internal permit udp host 192.168.129.20 any eq 123
access-list outbound-internal permit udp host 192.168.129.30 any eq 123
access-list outbound-internal permit tcp host 192.168.193.11 any eq 25
access-list outbound-internal permit tcp host 192.168.193.12 any eq 25
access-list outbound-internal permit tcp 192.168.0.0 255.255.0.0 host
      192.168.65.20 eq 3128
access-list outbound-internal permit tcp 192.168.0.0 255.255.0.0 host
      192.168.65.21 eq 3128
access-list outbound-internal permit 51 host 192.168.193.11 192.168.65.0
      255.255.255.0
access-list outbound-internal permit 51 host 192.168.193.14 192.168.65.0
      255.255.255.0
access-list outbound-internal permit 51 host 192.168.193.11 1.2.3.252
      255.255.255.254
access-list outbound-internal permit 51 host 192.168.193.14 1.2.3.252
      255.255.255.254
access-list outbound-internal permit 51 host 192.168.193.12 host 1.2.3.240
access-list outbound-internal permit tcp any 192.168.65.0 255.255.255.0 eq 22
access-list outbound-internal permit tcp any 192.168.65.0 255.255.255.0 eq 22
access-list outbound-internal permit tcp any 192.168.65.0 255.255.255.0 eq 80
access-list outbound-internal permit tcp any 192.168.65.0 255.255.255.0 eq 80
access-list outbound-internal permit tcp any 192.168.65.0 255.255.255.0 eq
      443
access-list outbound-internal permit tcp any 192.168.65.0 255.255.255.0 eq
      443
access-group inbound in interface external
access-group outbound-dmz in interface dmz
access-group outbound-internal in interface internal
pager lines 24
logging on
logging facility 19
logging buffered 4
logging console 5
logging host inside 192.168.193.14
logging trap 5
logging standby
mtu outside 1500
mtu inside 1500
mtu dmz 1500
mtu failover 1500
ip address outside 1.2.3.250 255.255.254.0
ip address inside 192.168.129.252 255.255.254.0
ip address dmz 192.168.65.252 255.255.255.0
ip address failover 192.168.254.1 255.255.255.0
ip local pool remote-vpn 192.168.253.10-192.168.253.59
```

```
ip audit info action alarm
ip audit attack action reset
fragment chain 1 outside
fragment chain 1 inside
fragment chain 1 dmz
arp timeout 14400
global (outside) 65  1.2.3.70-79   netmask 255.255.254.0
global (outside) 254 1.2.3.190-209 netmask 255.255.254.0
nat (inside) 193 192.168.193.0 255.255.255.0 0 0
nat (dmz) 65 192.168.65.0 19 0 0
nat (inside) 0 access-list ipsec-remote
static (dmz, outside) tcp 1.2.3.10 www 192.168.65.10 www netmask
      255.255.255.255
static (dmz, outside) tcp 1.2.3.11 443 192.168.65.11 443 netmask
      255.255.255.255
static (dmz, outside) tcp 1.2.3.20 smtp 192.168.65.20 smtp netmask
      255.255.255.255
static (dmz, outside) tcp 1.2.3.21 smtp 192.168.65.21 smtp netmask
      255.255.255.255
static (dmz, outside) tcp 1.2.3.50 443 192.168.65.50 443 netmask
      255.255.255.255
static (inside, outside) 1.2.3.241 192.168.193.11 netmask 255.255.255.255
static (inside, outside) 1.2.3.242 192.168.193.12 netmask 255.255.255.255
static (inside, outside) 1.2.3.244 192.168.193.14 netmask 255.255.255.255
route outside 0.0.0.0 0.0.0.0 1.2.3.1 1
timeout xlate 3:00:00
timeout conn 1:00:00 half-closed 0:10:00 udp 0:02:00 rpc 0:10:00 h225 1:00:00
timeout h323 0:05:00 mgcp 0:05:00 sip 0:30:00 sip_media 0:02:00
timeout uauth 0:05:00 absolute
aaa-server authenex protocol radius
aaa-server authenex host 192.168.193.15 GIACE_radius_key
aaa-server radius-acctport 1813
aaa-server radius-authport 1812
no http server enable
no snmp-server community public
no snmp-server community private
no snmp-server community cisco
no snmp-server community write
snmp-server location GIACE
snmp-server contact Security Administrator
snmp-server community GIACE_community
snmp-server host 192.168.193.11 trap
snmp-server host 192.168.193.11
snmp-server enable traps
floodguard enable
sysopt permit-ipsec
sysopt security fragguard
crypto ipsec transform-set internal ah-md5-hmac mode transport
crypto ipsec transform-set remote esp-aes esp-md5-hmac
crypto map internal 10 ipsec-isakmp
crypto map internal 10 match address ipsec-internal
crypto map internal 10 set transform-set internal
crypto map internal 10 set peer 192.168.193.0
crypto map internal 10 interface internal
crypto map remote 20 ipsec-isakmp dynamic dyn-remote
crypto map remote client token authentication authenex
crypto map remote client configuration address initiate
```

```
crypto map remote 20 interface outside
crypto dynamic-map dyn-remote 20 match address ipsec-remote
crypto dynamic-map dyn-remote 20 set transform-set remote
crypto dynamic-map dyn-remote 20 set pfs group2
isakmp enable outside
isakmp enable inside
isakmp policy 10 encryption aes
isakmp policy 10 group 2
isakmp key <strong_key1> address 0.0.0.0 netmask 0.0.0.0
isakmp key <strong_key2> address 192.168.193.0 netmask 255.255.255.0
vpngroup remote-ipsec address-pool remote-vpn
vpngroup remote-ipsec dns-server 192.168.129.20
vpngroup remote-ipsec dns-server 192.168.129.30
vpngroup remote-ipsec default-domain giace.com
vpngroup remote-ipsec split-tunnel ipsec-remote
vpngroup remote-ipsec idle-time 1800
vpngroup remote-ipsec max-time 43200
no telnet
no ssh
```

The standby firewall should receive the same configuration as above, with the exception of the following commands:

```
hostname GIACE-fw-b
ip address outside 1.2.3.251 255.255.254.0
ip address inside 192.168.129.253 255.255.254.0
ip address dmz 192.168.65.253 255.255.255.0
ip address failover 192.168.254.2 255.255.255.0
```

This will configure all features except for failover. To configure failover, power off the standby PIX to guard against any interference from the standby PIX. Then configure the failover addresses on the active PIX:

```
failover ip address outside 1.2.3.251
failover ip address inside 192.168.129.253
failover ip address dmz 192.168.65.253
failover ip address failover 192.168.254.2
failover mac address outside <act_mac> <standby_mac>
failover mac address inside <act_mac> <standby_mac>
failover mac address dmz <act_mac> <standby_mac>
no failover mac address failover
```

In the above commands, <act_mac> and <standby_mac> should be replaced with the MAC addresses for the appropriate active and standby interfaces. Note the “no failover mac address failover” command at the end. This is by default, but enabling MAC address failover on the LAN failover interface won’t work on a PIX, so we are explicitly ensuring that MAC failover is disabled on the LAN failover interface.

Next, activate stateful failover and LAN failover with the following:

```
failover link failover
failover lan interface failover
```

Tweak a couple of failover parameters:

```
failover poll 5
failover replicate http
```

The standby configuration is now in place on the active PIX. Configuration of the standby PIX is as simple as synchronizing the configuration by issuing the following on the active PIX:

```
write standby
```

This command forces the configuration to the standby PIX's running memory. One more command on the active PIX to ensure that the active PIX is the active firewall:

```
failover lan unit primary
```

At this point, both firewalls have the full configuration in the running memory. After firewall operation has been appropriately confirmed, the configurations should be saved to NVRAM so that the firewalls load the appropriate configuration after reboot. This is accomplished by issuing the following command on each PIX:

```
write memory
```

Firewall Policy Verification

No implementation should be considered complete until it is fully and properly verified. In the case of GIACE, the major verification consists of testing the border firewall to ensure that only anticipated traffic passes the firewall, although the border routers, internal routers, VPN and other security devices should be thoroughly tested as well. Some devices, such as the border and internal routers, are primarily traffic control devices and may be tested in much the same ways as detailed here. Other devices, such as the IDS nodes, perform in drastically different roles and will require appropriate testing methods. In this section, we will cover planning the test, the testing process as conducted, and evaluation of the results of the test.

Testing Plan

Testing a firewall implementation consists of transmitting a variety of traffic to the firewall and testing to determine what passes through. While this sounds very simple when starting to write a plan, the number of variations of legal and allowed, legal but denied, "gray" traffic (such as SYN/FIN packets), and illegal traffic means that many scenarios must be planned for and tested. As a result, a variety of tools are necessary. Unless otherwise specified, all tools run on the Linux platform and other Unix variants.

Here is a list and description of the tools used to test the GIACE firewall implementation:

- **tcpdump:** Tcpdump is an open-source packet- and frame-capture utility. (Windump is a similar package designed for Windows systems.) Tcpdump will be used on the testing side and the opposing side of the firewall to capture traffic traversing the firewall.
- **Nmap:** Nmap is an industry port-scanner and OS fingerprinting tool. Nmap will be used to find which ports are shown as open, closed, or no response (filtered). Nmap also has a Windows port. Nmap will report results to the best of its ability, however Nmap cannot determine the source of a particular response (if it was sent by the firewall, the host, an intermediate router, etc.). Therefore, all Nmap scan results will be verified by tcpdump.
- **Nessus:** Nessus is an open source vulnerability scanner. While Nessus is an extremely powerful tool, the primary use of Nessus will be to test for source-routed packets, various ICMP responses, and specific vulnerabilities against the firewall itself.

These tools, in combination, provide a powerful method of validating any traffic-filtering device, such as a firewall.

In general, the test shall consist of generating traffic (stimulus) and recording the behavior evoked by the traffic in question (response). The response is then confirmed to be an appropriate response for the associated stimulus.

Traffic conditions to be tested include:

- TCP SYN for all possible ports
- UDP traffic to all possible ports
- Normal fragmentation
- Overlapping fragmentation
- Unusual/Illegal TCP flags
  - SYN/FIN
  - SYN/RST
  - Null
  - FIN/PSH/URG (Christmas Tree)
- Source-routed packets

Most tests are only of concern for traffic arriving from the Internet, however the TCP and UDP tests will be repeated along every relevant traffic flow:

- Internet -> GIACE hosts. Specifically:
  - Web Servers
  - E-Commerce Servers
  - Proxy/Email Gateways
  - Logging Server
  - SNMP Server
- DMZ -> Internet
- DMZ -> Internal Network
- Internal Hosts -> Internet. One typical host plus:
  - Logging Workstation
  - SNMP Workstation
  - IDS Management Workstation

In each part of the test, GIACE will connect a testing machine (the “traffic generator”) with the appropriate software installed to the appropriate source network in the same manner as a typical host on that segment. Another test machine (the “sniffer”) will be

connected to the target network to a switch port configured to mirror the firewall port so that all traffic traversing the firewall will also be available to the sniffer.

Denial-of-Service (DoS) tests such as Teardrop, Fraggle, SYN Flooding, etc. will not be performed due to the potential service interruption. If GIACE wishes for DoS tests to be performed in the future, the tests should occur during a maintenance window so as to impact the operation of GIACE as minimally as possible.

The test plan is not expensive from a monetary standpoint, however, it will be somewhat labor intensive to perform. Testing may occur at any time without significantly impacting GIACE operations, subject to the DoS test caveat mentioned above. GIACE may wish to restrict testing to off-hours or maintenance windows to ensure high availability if a test creates unexpected results.

Remember, this test is meant to be a validation of the GIACE firewall, not a full vulnerability or penetration test. While it is impossible to test only the firewall due to the multiple layers of security, all attempts will be made to verify the firewall individually in this test. Therefore, the results are only useful when applied to the firewall, and should be given only appropriate weight when applied to the entire GIACE network. GIACE should have further testing performed in the future to ensure that the GIACE network as a whole is appropriately secured.

Conducting the Test and Result Analysis

Firewall testing occurs in several phases: basic TCP port scanning, basic UDP port scanning, unusual TCP flag port scanning, normal fragmentation, overlapping fragmentation, and source-routed packets. Each will be dealt with individually in the following subsections. Some of these phases may overlap to save time, however care should be taken to ensure that only one host is tested at a time by any single test to prevent confusion in the tcpdump captures. For the purposes of this test, all testing from the Internet will originate from 1.2.3.200. This is a host inside the border router, but outside the firewall. This enables the test to test the firewall only, without effect from intermediate Internet routers or the GIACE border routers.

Port Scanning

Basic TCP port scanning is a staple of almost any security testing methodology. Simply, TCP port scanning consists of sending TCP SYN packets to a range of ports. If the target host responds with a TCP SYN/ACK packet, the port is open. If the target host responds with TCP RST, the port is closed. If no response is received, the test is inconclusive, since the lack of response may be due to a firewall or other traffic control device or other network conditions. Two basic variations on the TCP port scan exist: a connect scan and a “stealth” scan. The only difference between the two is that when the connect scan receives a SYN/ACK in return, the connect scan completes the connection with an ACK and immediately tears down the connection. The “stealth” scan

instead sends an RST, helping to evade logging by dumber/older systems and lessening the traffic load. In our tests, we will use a “stealth” scan due to its wider use.

To initiate capturing for this test, capture nodes on the internal and DMZ segments will run tcpdump as follows:

```
tcpdump -ns 0 host 1.2.3.200
```

This command starts tcpdump without DNS support (-n), tells it to capture the entire length of the frame (-s 0), and only captures traffic to or from IP 1.2.3.200 (host 1.2.3.200). Other options which may be used include “most verbose” mode (-vvv), display captures in hex (-x) and display with ASCII interpretations (-X). Basically, if all the options presented here are used, tcpdump will display as much information as possible about the captured frame, but will strip all traffic that isn’t related to our testing host.

A port scan is initiated directly against the firewall with the following:

```
nmap -sS -p 1-65535 -v -v -P0 1.2.3.[250,251]
```

Nmap will use a stealth scan (-Ss), scan all ports (-p 1-65535), report as verbosely as possible (-v -v), not ping hosts to determine whether they are up (-P0), and scan hosts 1.2.3.250, and 1.2.3.251. These IP addresses are assigned to the two PIX firewalls.

Since there are no ports open from the outside of the firewall, results from Nmap look like this:

```
Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Host 1.2.3.250 appears to be up ... good.
Initiating SYN Stealth Scan against 1.2.3.250
The SYN Stealth Scan took 5446 seconds to scan 65535 ports.
All 65535 scanned ports on 1.2.3.250 are: filtered

Host 1.2.3.251 appears to be up ... good.
Initiating SYN Stealth Scan against 1.2.3.251
The SYN Stealth Scan took 5430 seconds to scan 65535 ports.
All 65535 scanned ports on 1.2.3.251 are: filtered

Nmap run completed -- 2 IP addresses (2 hosts up) scanned in 10876 seconds
```

No ports were found to be open by Nmap, as desired. Notice that scanning all 65535 ports took quite some time. The firewalls are configured to drop all traffic, so Nmap must wait for each port attempt to timeout in order to conclude that the traffic is filtered. Next, we check the tcpdump results to ensure that it is the firewall providing the expected results, not some other configuration, by checking the output from tcpdump. Both the DMZ and the internal tcpdump results look similar to the following:

```
# tcpdump -ns 0 host 1.2.3.200
tcpdump: listening on eth0
```

```
1 packets received by filter
0 packets dropped by kernel
```

The first line shows tcpdump starting and specifies on which interface tcpdump is listening. The last two lines show a final packet count as tcpdump terminates.[1] Since there is no output in between, no traffic has been received. Since tcpdump was started with a filter to specifically capture traffic from the testing station, we may conclude that no traffic has passed the firewall. Hereafter, results similar to these will be referred to as an "empty tcpdump capture."

The tcpdump results show that no traffic was allowed to cross the firewall. This is the desired result.

This test is then repeated for the other, producing the following Nmap results (only web servers shown):

```
#nmap -sS -p 1-65535 -v -v -P0 1.2.3.[10,11,50]
Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Host 1.2.3.10 appears to be up ... good.
Initiating SYN Stealth Scan against 1.2.3.10
Adding open port 80/tcp
The SYN Stealth Scan took 5448 seconds to scan 65535 ports.
Interesting ports on 1.2.3.10:
(The 65534 ports scanned but not shown below are in state: filtered)
Port       State       Service
80/tcp     open        http

Host 1.2.3.11 appears to be up ... good.
Initiating SYN Stealth Scan against 1.2.3.11
Adding open port 443/tcp
The SYN Stealth Scan took 5433 seconds to scan 65535 ports.
Interesting ports on 1.2.3.11:
(The 65534 ports scanned but not shown below are in state: filtered)
Port       State       Service
443/tcp     open        https

Host 1.2.3.50 appears to be up ... good.
Initiating SYN Stealth Scan against 1.2.3.50
Adding open port 443/tcp
The SYN Stealth Scan took 5437 seconds to scan 65535 ports.
Interesting ports on 1.2.3.50:
(The 65534 ports scanned but not shown below are in state: filtered)
Port       State       Service
443/tcp     open        https

Nmap run completed -- 3 IP addresses (3 hosts up) scanned in 16318 seconds
```

[1] The packet counts are dependent on operating system and kernel configuration and therefore should not be given much weight unless detailed operating system information is known. It appears that the counter shows 1 on this platform if no packets arrive; otherwise the packet count is the number of packets that match the filter. Other platforms may show the number of packets that arrive at the interface prior to filtering. For more details, see the tcpdump man page.

This time we show one open port per IP address, corresponding to the appropriate HTTP(S) port allowed for each server. The tcpdump output will be different as well (shown for only one host):

```
# tcpdump -ns 0 host 1.2.3.200
tcpdump: listening on eth0
09:33:35.650483 1.2.3.200.46687 > 192.168.65.10.80: S 174538387:174538387(0)
      win 3072
09:33:35.650597 192.168.65.10.80 > 1.2.3.200.46687: S
      1107840400:1107840400(0) ack 174538388 win 5840 <mss 1460> (DF)
09:33:35.653684 1.2.3.200.46687 > 192.168.65.10.22: R 174538388:174538388(0)
      win 0
```

As the tcpdump output shows, a SYN from the testing host arrives on the DMZ side of the firewall, the target host sends a SYN/ACK, and the testing host immediately responds with an RST. This shows that client traffic will be allowed to build a TCP connection through the firewall. Since these services are allowed on the respective web servers, this is the expected result.

The port scan is then repeated for UDP ports on all hosts by substituting the –sS option with –sU. Nmap performs UDP scans by sending a packet to the respective port. If an ICMP Port Unreachable message is returned, the port is closed. If no response is returned, Nmap assumes the port is open. This often makes UDP port scanning through firewalls very inaccurate, so it is critical that this test is verified with tcpdump. Since no UDP traffic is allowed through the firewall unless an entry is already present in the state table, an empty capture results. This is the expected behavior.

Nmap is run several times more to detect responses to unusual TCP flag combinations:

```
#nmap -sF -p 1-65535 -v -v -P0 1.2.3.[10,11,20,21,50,250,251]
#nmap -sX -p 1-65535 -v -v -P0 1.2.3.[10,11,20,21,50,250,251]
#nmap -sN -p 1-65535 -v -v -P0 1.2.3.[10,11,20,21,50,250,251]
```

The above commands scan each configured IP address for responses to FIN packets, Christmas Tree packets (FIN/PSH/URG flags set), and Null packets (no flags). FIN packets are significant since they are normal traffic, but when sent to a host port that does not already have a session established, the response varies when the port is open as opposed to when it is closed.[1] Christmas Tree packets are just an unusual flag combination. Null packets should not normally be seen, as all TCP packets normally carry the at least either the SYN, ACK, RST, or FIN flag, if nothing else. None of these combinations should be allowed through the firewall, so tcpdump reveals an empty capture.

Nmap performs one last test of interest: an IP Protocol scan. This scan sends packets of various IP protocols to the target in an attempt to elicit an ICMP Protocol Unreachable message. If an ICMP Protocol Unreachable is received, the host does not

[1] TCP/IP spec states that closed ports should respond with RST. Open ports should not respond to the FIN packet. Microsoft systems are broken however and not susceptible to FIN scanning.

understand that protocol. If no response is recieved, the protocol is likely active on the host. The firewall suppresses ICMP Protocol Unreachable messages, so all protocols appear to be available in the scan:

```
Interesting protocols on  (1.2.3.250):
Protocol   State       Name
1          open        icmp
2          open        igmp
3          open        ggp
4          open        ip
5          open        st
6          open        tcp
7          open        cbt
8          open        egp
9          open        igp
10         open        bbn-rcc-mon
11         open        nvp-ii
12         open        pup
13         open        argus
14         open        emcon
15         open        xnet
16         open        chaos
17         open        udp
...
249        open        unknown
250        open        unknown
251        open        unknown
252        open        unknown
253        open        unknown
254        open        unknown

Nmap run completed -- 1 IP address (1 host up) scanned in 323 seconds
```

As mentioned above, the TCP and UDP port scans should be repeated for DMZ and outbound traffic flows as well. The more unusual scans may be omitted to save time.

Fragmentation Tests

Fragmentation tests will also be initially performed using Nmap by using the –f switch. This switch tells Nmap to perform the scan with packets fragmented into small chunks. This tests whether or not fragments will be allowed to pass the firewall, and whether they are reassembled in the process. Since we do not allow fragmented packets to pass the firewall, no fragments should be seen in a tcpdump capture. Therefore, the test looks exactly like a normal portscan when captured:

```
# tcpdump -ns 0 host 1.2.3.200
tcpdump: listening on eth0
09:33:35.650483 1.2.3.200.46687 > 192.168.65.10.80: S 174538387:174538387(0)
      win 3072
09:33:35.650597 192.168.65.10.80 > 1.2.3.200.46687: S
      1107840400:1107840400(0) ack 174538388 win 5840 <mss 1460> (DF)
09:33:35.653684 1.2.3.200.46687 > 192.168.65.10.80: R 174538388:174538388(0)
      win 0
```

Vulnerability Scan/Source Routed Packets

Nessus tests for a wide variety of vulnerabilities, but the ones of most interest are for source routed packets and ICMP response leakage. By using Nessus for this part of the test, we also get the added benefit of detecting other vulnerabilities in the PIX OS software that have not been patched. As expected, the scan reveals no vulnerabilities:

Summary of scanned hosts

Host	Holes	Warnings	Open ports	State
1.2.3.250	0	0	0	Finished
1.2.3.251	0	0	0	Finished

Recommendations

The results presented above show a strong defense for the GIACE network that is consistent with the firewall policy. Since only the firewall was tested, however, additional testing may be desired. End-to-end testing is the best way to determine compliance with the overall perimeter policy. Also, it may be desirable to test how the firewalls reassemble overlaid fragments, since the above testing only proves that fragments will be reassembled at the firewall and does not demonstrate how they are reassembled. [1] Testing similar to the firewall test should occur for every traffic control point; in GIACE's case this would consist of border routers and internal routers. Overall, a complete vulnerability assessment and penetration test should be performed to gauge the status of the GIACE perimeter as a whole and evaluate the effectiveness of the IDS, logging and alerting systems.

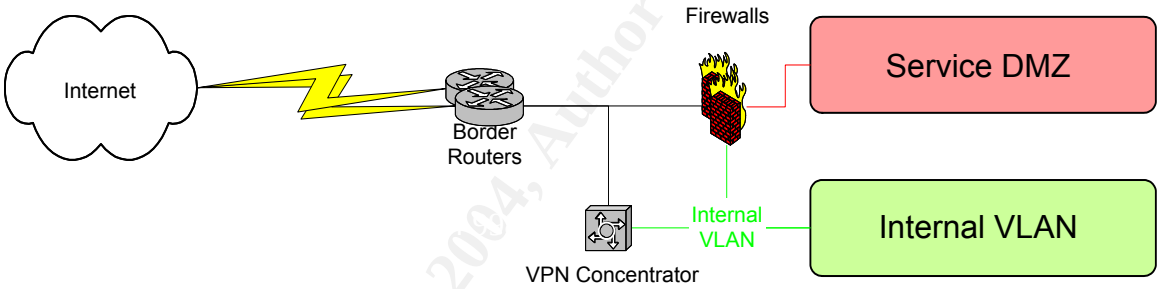
Other recommendations are as follows:

- Weaknesses in the perimeter are primarily based around availability. Installation of a backup authentication server would ensure that VPN is always available. This may be especially important since VPN will be used for remote support during off hours. GIACE has chosen to accept the risk of VPN downtime for the time being however. Similarly, the logging server, IDS console, and network management console are all single points of failure.
- The network IDS node's effectiveness in the DMZ is somewhat weakened by the use of HTTPS all the way to the web server for the E-Commerce servers. This could be improved by installation of an SSL Accelerator or SSL-enabled reverse-proxy

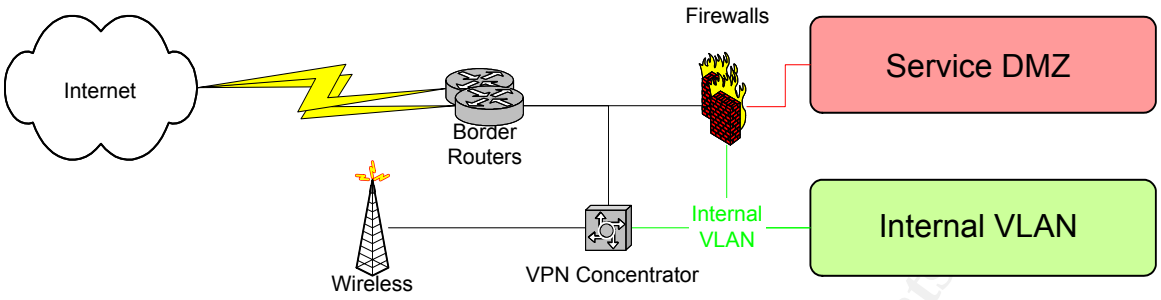
[1] Overlaid fragments are fragmented IP packets in which data from one packet overlaps with another. This causes some data to be lost and is useful for disguising malicious activity from IDS. Familiarity with the way the firewalls reassemble these packets will assist with IDS activity analysis.

system in front of each server. The reverse-proxy would also have the added benefit of greater protocol inspection for the E-Commerce traffic.

- Similarly, the IDS may or may not be effective against IPSec-AH traffic as it passes to the administrative hosts, depending on how the IDS rules are written. Testing, and re-writing of the IDS rules if necessary, are recommended to ensure that attacks against the administrative hosts are not hidden through encapsulation within AH traffic.
- Network traffic may oversubscribe some links in the future. IDS links are particularly at risk, and Cisco recommends monitoring LAN Failover enabled links for bandwidth utilization, particularly when stateful failover is enabled. Careful network monitoring will identify such conditions, allowing for time to consider upgrades to the appropriate interfaces.
- Load on the firewalls may prove to be undesirable over time as the number of VPN users grows. If this becomes of concern, a VPN concentrator device may be installed. If a concentrator is installed in the future, the following configuration will provide the easiest installation without providing change in functionality:



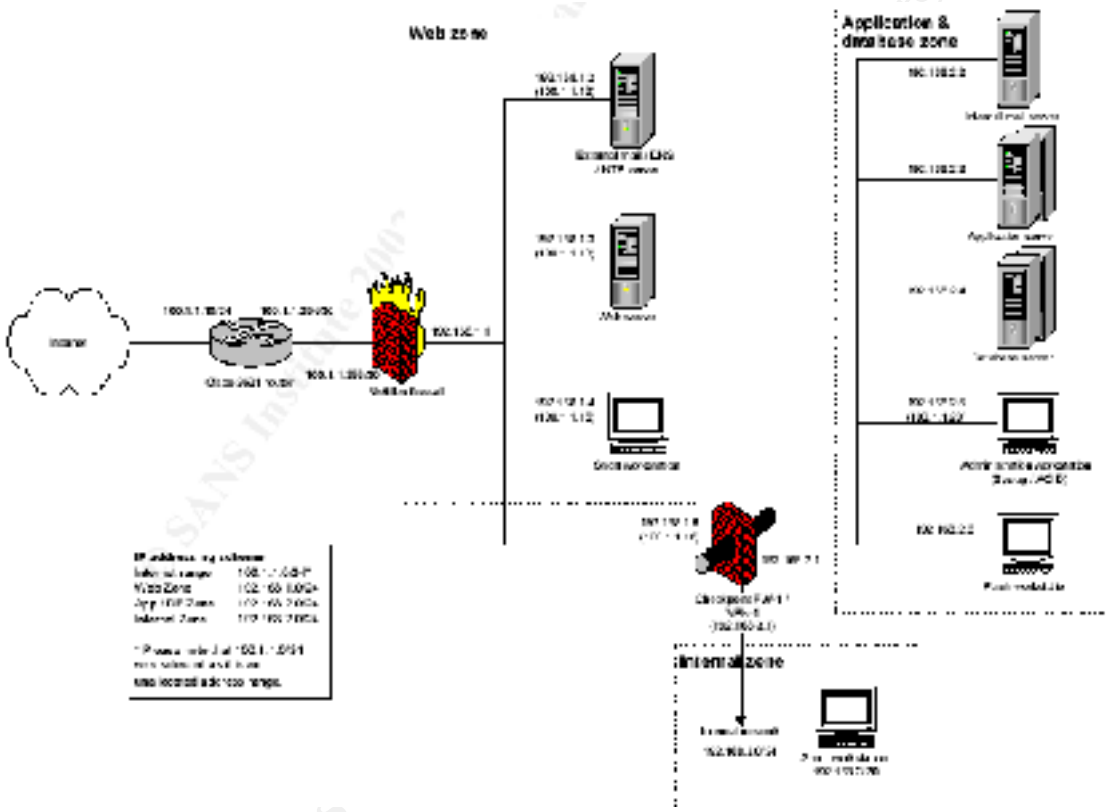
- Currently, all VPN users are trusted, and traffic is not controlled between VPN users and the GIACE network. This presents high risk of damage if the VPN were to become compromised. GIACE may wish to limit VPN traffic in the future after performing a traffic analysis to determine typical usage of the VPN. If this is performed, GIACE may with to implement a VPN concentrator in a different configuration than demonstrated above to better facilitate the needs of the VPN.
- GIACE may wish to implement wireless in the future. Wireless is a tricky platform to secure and therefore wireless should be implemented behind a traffic control device of some kind. One possible implementation method could be to attach the wireless network to an additional interface on the VPN device (either a concentrator or the firewall, depending on if GIACE has implemented a concentrator or not) and requiring all users to authenticate to the VPN before allowing access to the GIACE network. Wireless technology is currently developing rapidly. Therefore, GIACE should evaluate all available solutions at the time of implementation if wireless is implemented in the future. The following diagram shows a possible Wireless configuration after a VPN Concentrator implementation:



- Host firewalls could be deployed on all workstations to provide the greatest protection to the internal network. Host firewalls are already deployed on DMZ nodes due to the risk of residing in the DMZ and the potential access to company data. Deployment on the internal network would provide granular traffic control and visibility on the Internal network as well.

Design Under Fire

Design under fire is an opportunity to provide peer review to other designs and demonstrate that perimeter security devices provide only a piece in the security puzzle. Three parts make up this section: an attack against the firewall, a Distributed Denial of Service attack (DDoS), and an attack against an internal system. The attacks will be in context of this network design taken from Timothy Miller's practical:[1]



Attack Against the Firewall

The frontline firewall for this design is a Redhat Linux 8.0 system upgraded to NetFilter 1.2.8. (RH8 ships with 1.2.7a by default.) SSH is probably enabled on this system since this system is secured using the CIS Linux Benchmark,[2] which specifically allows SSH, and Miller did not mention disabling it in his paper, but we will come back to that.

Linux firewall features perform as part of three tables: Input, Output and Forward. These tables are relatively self explanatory, the only catch is that Forward table in the 2.4 kernel only processes packets which would be forwarded by the host, and Input and Output only process packets arriving at the host and leaving the host respectively.

[1] Miller, http://www.giac.org/practical/GCFW/Timothy_Miller_GCFW.pdf.
[2] Miller did not specify the version of the benchmark he used. This paper assumes he was using version 1.1.0, the latest as of writing of this paper.

Users may define additional tables, but user-defined tables must be called from one of the three default tables. Therefore, since the attack is directed at the firewall itself, we can ignore the Forward and Output tables and concentrate on the Input chain for now. Miller defines the Input tables as follows:

```
$IPTABLES -A INPUT -p tcp -j bad_tcp_packets
$IPTABLES -A INPUT -p ALL -i $LO_IFACE -s $LO_IP -j ACCEPT
$IPTABLES -A INPUT -p ALL -i $LO_IFACE -s $INET_IP -j ACCEPT
$IPTABLES -A INPUT -i $INET_IFACE -j spoofed_packets
$IPTABLES -A INPUT -p ALL -d $INET_IP -m state -state
   ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A INPUT -m limit -limit 3/minute --limit-burst 3 -j LOG --
   log-level DEBUG --log-prefix "IPT INPUT packet died: " [1]
```

The input table references two user-defined tables. "bad_tcp_packets" is a table that strictly logs and drops TCP packets that do not have an entry in the state table, and is fairly straightforward. "spoofed_packets" is supposed to be a table to drop spoofed packets:

```
$IPTABLES -N spoofed_packets
$IPTABLES -A spoofed_packets -i $INET_IFACE -s $GIAC_RANGE -j DROP
$IPTABLES -A spoofed_packets -i $INET_IFACE -s $CLASS_A -j DROP
$IPTABLES -A spoofed_packets -i $INET_IFACE -s $CLASS_B -j DROP
$IPTABLES -A spoofed_packets -i $INET_IFACE -s $CLASS_C -j DROP
$IPTABLES -A spoofed_packets -i $INET_IFACE -s $CLASS_D_MC -j DROP
$IPTABLES -A spoofed_packets -i $INET_IFACE -s $CLASS_E_RES -j DROP [2]
```

The variable $GIAC_RANGE is defined as the GIAC public IP range; the CLASS_* variables contain the reserved/private IP address spaces. The problem with this table is that it does not take into account destination addresses. Normally RFC-1918 addresses are not supposed to be routable on the Internet. Misconfigured routers will sometimes respect source-routed packets however, possibly giving the ability to get an RFC-1918 destined packet to the firewall host. If the packet is source routed to the firewall, and the destination address is the RFC-1918 address assigned to the internal interface of the firewall (192.168.1.1), the firewall will not filter the packet, and the Linux kernel will happily accept the packet and process it since it is addressed to an IP address assigned to a local interface.

If SSH is enabled on the firewall (which it probably is), an attack can be performed against SSH. A noisy and crude attack would consist of guessing or social engineering a login name and then brute forcing the password. RedHat 8 shipped with OpenSSH version 3.4p1, which allows an attacker to use the OpenSSH/PAM timing attack[3] to accelerate discovery of a username, however this will still be slow and noisy. A more elegant attack for an experienced attacker would be to develop an exploit for the

[1] Miller, http://www.giac.org/practical/GCFW/Timothy_Miller_GCFW.pdf.
[2] Miller, http://www.giac.org/practical/GCFW/Timothy_Miller_GCFW.pdf.
[3] Several bulletins for this issue exist, however the best available is probably from ISS: http://xforce.iss.net/xforce/xfdb/11970. Full citation is contained in the Bibliography.

OpenSSH Large Packet Buffer Overflow[1] that incorporates source routing and a backchannel of some sort. Unless the exploit is developed to manipulate the firewall rules directly (difficult!), the backchannel must be allowed by the current firewall rules. Since we have already circumvented the rules using source routing, the same trick is probably the most viable for the back channel: open a shell port on the 192.168.1.1 address and use source route to access it, at least long enough to manipulate the firewall rules to something more direct.

This is an above-average difficulty attack; however it is very viable if source routing is not disabled and the border router does not explicitly filter against RFC-1918 addresses. Luckily for GIAC, this is the case on the border router for this design, so the attack would fail. The firewall could better guard against this attack by filtering against the unused RFC-1918 addresses in all directions, not just inbound on the external interface.

Distributed Denial of Service (DDoS) Attack

DDoS attacks are the subject of great debate across the Internet today due to the difficulty in counteracting attacks. DDoS attacks consist of using a variety of hosts at varying locations to create traffic that consumes resources such as bandwidth, memory, state table space, or any other finite resource. Most DDoS attacks are based from a group of Internet connected hosts that have been compromised by an attacker or worm. These hosts have remotely controllable software installed, and are known as "zombies" due to these remote control abilities without the knowledge of the host owner. Zombie networks of 50 systems or greater are not difficult to accomplish, especially given the poor security record of common operating systems and frequent slow patch response. The latest widespread security issue is the Microsoft MDAC unchecked buffer detailed in Microsoft bulletin MS03-033.[2] Several exploits exist in the wild, enabling the installation of any number of backdoor software packages. While the Trinoo tool is a relatively well-known tool, it is still commonly used in the wild and very flexible.

Miller's design does not specify the bandwidth on the Internet connection, however it does specifically show a single router and connection to the Internet. Therefore, it is logical that the connection is at most a single T1 line, and possibly even a fractional T1 line. After compromise and installation of the 50 remote hosts, the Trinoo tool will be used to attack Miller's GIACE network with a variety of bogus UDP packets sent to random ports. The border router ACLs will generally block these packets; however, the damage will have already been done, since the UDP packets will have already crossed the T1 line before filtering. Each zombie would need to generate only four packets per second at 1000 bytes per packet to effectively saturate a single T1.[3] This packet rate can easily be accomplished over even a 33.6K dialup connection. With the prevalence

---

[1] Again, there are several bulletins, but the best reference is from ISS: http://xforce.iss.net/xforce/xfdb/13191.

[2] The entire bulletin may be located at: http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS03-033.asp

[3] 4 pps x 50 hosts = 200 pps, 200pps x 1000 B = 200,000 B/sec., 200 KB/sec x 8 bits/byte = 1.6 Mbps. A T1 performs at 1.544 or 1.536 Mbps depending on configuration.

of high-bandwidth broadband connections, not all 50 hosts will be restricted to 33.6K of bandwidth, and will easily overrun a single T1.

Extra bandwidth is often cited as a countermeasure against bandwidth-based DDoS attacks, however, it is often not the most effective countermeasure. Even with the relatively small number of hosts employed in this attack, 50 dialup zombies is still sufficient to overrun multiple T1s, not to mention what is possible with broadband connected zombies. Many administrators react by contacting their upstream ISP to request ACL filtering against the offending IP addresses. This does not scale well, especially given the prevalence of dynamic IP addresses on the Internet. (Broadband IP addresses change regularly, and dialup addresses change with every reconnection). A more effective countermeasure would be to filter inbound traffic at the upstream ISP to only explicitly allowed UDP traffic. The problem with this method is that certain UDP traffic is almost always necessary (such as UDP/53 – DNS), and if an attacker limits traffic to this port, it will pass the filter.

The most effective control method consists of bandwidth shaping on the upstream ISP router to limit UDP traffic to only a certain amount of bandwidth. Therefore, a UDP based attack will consume a certain amount of bandwidth on the T1 (say 200K), but will be unable to consume the entire T1. More granular shaping can be used to ensure that even UDP traffic from certain hosts (like the upstream NTP servers) is preferred and will pass the T1 line first. Of course, this relies on careful coordination with a knowledgeable and willing ISP.

Attack Against an Internal System

The database server usually contains the most valuable data in an organization, and consequently is the target of this attack. Like most E-Commerce configurations, Miller has placed a server to provide the web front-end for the system in a service DMZ, and required the web front-end to pass through a firewall when communicating with the database backend. This allows GIACE the opportunity to prevent Internet hosts from ever communicating directly with the database server, instead routing all communication through a program that parses the data. This provides significant protection for the database server; however, flaws in the web program may still be exploited to attack the database server. Web injection attacks are currently popular and effective when web programming has not been subjected to appropriate code review and testing steps before installation in a production environment. This attack does not occur at the Transport Layer, but instead at the Application Layer.[1] Attacks at the Application Layer may quite effectively circumvent lower layer security measures, however, as we will soon see.

Before we begin, while Miller specified DB2 as the database software of choice, he did not specify an operating system for the database servers. Since Miller has chosen to

[1] The TCP/IP Layer model is used here instead of the OSI model. The Transport TCP/IP Layer roughly translates to the Transport Layer in the OSI model, however the Application Layer in the TCP/IP model translates roughly to all layers above the Transport OSI Layer.

use RedHat 8.0 extensively in his design, it is reasonable to assume that the DB2 servers are also RedHat, and that assumption is made for the rest of this section.

Injection attacks consist of providing unexpected input to a web form. If the web application does not properly parse out code from the input, the input may be passed to the web server which then may interpret and execute the code. For example, inserting:

```
' or 1=1--
```

into a username field may bypass login credential entirely, or at least provide an error revealing usernames, table layouts, or other critical information. In other cases, systems calls may be made, allowing attackers to compromise login tables, open backdoor shell access, or offloading data directly to unauthorized hosts. Full treatment of SQL injection is beyond the scope of this document, however several excellent references are available on the web.[1] The important concept is that the entire system may be compromised by an attacker taking advantage of improper input validation.

Several methods exist to counter injection attacks. The most obvious is proper input validation in the web application. For example, all input should be processed to a standard encoding method to remove Unicode encoding based methods. Also, special characters such as quotes (“), hyphens (-), semicolons (;) and other punctuation should be stripped out unless specifically acceptable in a field. If they are acceptable in a particular field, they should be encoded in such a way that they cannot be parsed by the database.

The database server configuration may provide additional mitigation. Running the database service in a restricted security context will help contain what damage may be sustained from successful attack. Restricting access to certain procedures, such as system calls, will also help to contain what damage an attacker may cause. The database software vendor (IBM in this case) should be able to provide additional recommendations for software configuration steps.

The final category for counter-measures to prevent against injection attacks consists of filtering HTTP input before processing by the web server. Apache provides mechanisms for this, and Miller has provided some configuration along these lines in his router configuration, however the filters provided on the router will not be very effective. First, the URLs the router is filtering against are not well tuned for Apache servers, and contain a large number of filters against IIS attacks. More importantly, the forms susceptible for injection attacks are likely to be protected by HTTPS, which means that the router will not be able to examine the packet payloads. Therefore, this filtering should occur on the web server itself, such as by using the mod_rewrite and other input filtering and modification features of Apache.

[1] One such reference is available at http://seclists.org/lists/webappsec/2003/Jul-Sep/att-0128/DB2_database_mining_with_SQL_injection.doc.

Appendix A: IP Address Assignments

IP	Device	Hostname	Purpose
1.2.3.1	VIP	giace-gw	HSRP Default Gateway
1.2.3.2	VIP	giace-fw	HSRP Default Gateway
1.2.3.10	NAT	www	NAT for Informational Website
1.2.3.11	NAT	ecom	NAT for E-Commerce Website
1.2.3.20	NAT	blackhole	NAT for Proxy Server
1.2.3.21	NAT	wormhole	NAT for Proxy Server
1.2.3.50	NAT	ca	NAT for Certificate Authority
1.2.3.240	IDS	ext-ids	External IDS sensor
1.2.3.250	PIX Firewall	giace-fw-a	Firewall
1.2.3.251	PIX Firewall	giace-fw-b	Firewall
1.2.3.252	Router	giace-gw-a	Border Router
1.2.3.253	Router	giace-gw-b	Border Router
192.168.1.1	VIP	db-gw	HSRP Default Gateway
192.168.1.11	Workstation	lotto	Lottery Number Generator
192.168.1.20	DB Server	db-fortune-a	Fortune Database
192.168.1.21	DB Server	db-fortune-b	Fortune Database
192.168.1.40	DB Server	db-crm-a	Customer Database
192.168.1.41	DB Server	db-crm-b	Customer Database
192.168.1.252	MSFC	db-gw-a	Routing Card
192.168.1.253	MSFC	db-gw-b	Routing Card
192.168.65.1	VIP	dmz-gw	HSRP Default Gateway
192.168.65.11	VIP	www	Informational Website Virtual IP
192.168.65.12	VIP	ecom	E-Commerce Website Virtual

IP			
192.168.65.20	Proxy Server	blackhole	Proxy/Mail Relay
192.168.65.21	Proxy Server	wormhole	Proxy/Mail Relay
192.168.65.30	Web Server	www-a	Informational Website
192.168.65.31	Web Server	www-b	Informational Website
192.168.65.40	Web Server	ecom-a	E-Commerce Website
192.168.65.41	Web Server	ecom-b	E-Commerce Website
192.168.65.50	CA Server	ca	Certificate Authority
192.168.65.240	IDS Sensor	dmz-ids	DMZ IDS Sensor
192.168.65.250	Switch	dmz-sw-a	Switch
192.168.65.251	Switch	dmz-sw-b	Switch
192.168.65.252	PIX Firewall	gw-a	Firewall
192.168.65.253	PIX Firewall	gw-b	Firewall
192.168.129.1	VIP	int-fw	HSRP Internet Gateway
192.168.129.10	Terminal Server	ts	Terminal Server
192.168.129.20	File Server	cabinet	File/Print Server
192.168.129.30	Groupware Server	postman	Exchange Server
192.168.129.50-240	Workstations		
192.168.129.252	PIX Firewall	int-fw-a	Firewall
192.168.129.253	PIX Firewall	int-fw-b	Firewall
192.168.130.1	VIP	int-gw	Routing Card - Default Gateway
192.168.130.10-149	Workstations		
192.168.130.150-199	VPN Clients		
192.168.130.253	MSFC	int-gw-a	Routing Card
192.168.130.254	MSFC	int-gw-b	Routing Card

192.168.193.1	VIP	admin-gw	HSRP Default Gateway
192.168.193.11	Workstation	siren	Network Monitoring Workstation
192.168.193.12	Workstation	tripwire	IDS Monitoring Workstation
192.168.193.13	Backup Server	vault	Backup Server
192.168.193.14	Logging Server	papyrus	Logging Server
192.168.193.15	Token Authent.	two-bits	Token Authentication Server
192.168.193.240	IDS Sensor	int-ids	Internal IDS Sensor
192.168.193.253	MSFC	admin-gw-a	Routing Card
192.168.193.254	MSFC	admin-gw-b	Routing Card

Appendix B: Sample Border Router Config

The following is a completed configuration file for one of the Cisco 3640 border routers. This file is meant to be transferred to the router by TFTP on initial configuration, and therefore has additional commands and comments that will not necessarily be present in the "show running-config" output after the router is configured. Cisco does not store comments in router online configurations, and some of the commands are reinforcing defaults in case it becomes necessary to load the configuration on a version of IOS other than 12.3.

```
! Anti-congestion algoritm
service nagle
no service pad
! Verbose logging timestamps
service timestamps debug datetime msec localtime show-timezone
service timestamps log datetime msec localtime show-timezone
! Encrypt all passwords
service password-encryption
no service dhcp
!
hostname GIACE-bdr-1
!
logging buffered 16384 notifications
! Crackers get a life!  Stronger passwords will be used in the final config.
enable secret 5 $1$lTyH$GCuhuZK3agVQtfGZ5maJj1
!
username GIACEuser password 7 104D000A0618
!
clock timezone EST -5
ip subnet-zero
no ip source-route
no ip finger
no ip domain-lookup
!
no ip bootp server
!
interface Null0
 no ip unreachables
!
interface Serial0/0
 description T1 to ISP
! Share the IP with the ethernet interface - reconfig if the ISP doesn't do
      unnumbered
 ip unnumbered FastEthernet0
 ip access-group antispoof-ingress in
 ip access-group antinoise out
! don't send ICMP errors or proxy-arp
 no ip redirects
 no ip unreachables
 no ip proxy-arp
! keep statistics on access list violations
 ip accounting access-violations
```

```
 ip multicast boundary 30
 rate-limit input access-group 150 200000 25000 25000 conform-action transmit
      exceed-action drop
 rate-limit input access-group 160 50000 6250 6250 conform-action transmit
      exceed-action drop
 ip route-cache flow
 shutdown
 no cdp enable
!
interface FastEthernet0/0
 description Untrusted Segment Interface
 ip address 1.2.3.253 255.255.254.0
 ip access-group antispoof-egress in
 ip access-group antinoise out
 no ip redirects
 no ip unreachables
 no ip proxy-arp
 ip accounting access-violations
! do not pass multicast
 ip multicast boundary 30
 ip route-cache flow
 speed auto
 arp timeout 0
 no cdp enable
! enable HSRP for failover
 standby authentication GIACE-ro
 standby name GIACE-bdr-gw
 standby ip 1.2.3.1
 standby track Serial0
!
! Configuration of BGP and routing protocols is beyond scope and ISP
      dependent
router bgp 65533
!
ip classless
! Default route - change with ISP info
ip route 0.0.0.0 0.0.0.0 1.1.1.1
! null routes for bogus stuff (just in case, should be killed by the ACL as
      well)
! may want to aggregate these in the future to save memory, but will be
      harder to maintain
ip route 2.0.0.0 255.0.0.0 Null0
ip route 5.0.0.0 255.0.0.0 Null0
ip route 7.0.0.0 255.0.0.0 Null0
ip route 10.0.0.0 255.0.0.0 Null0
ip route 23.0.0.0 255.0.0.0 Null0
ip route 27.0.0.0 255.0.0.0 Null0
ip route 31.0.0.0 255.0.0.0 Null0
ip route 36.0.0.0 255.0.0.0 Null0
ip route 37.0.0.0 255.0.0.0 Null0
ip route 39.0.0.0 255.0.0.0 Null0
ip route 41.0.0.0 255.0.0.0 Null0
ip route 42.0.0.0 255.0.0.0 Null0
ip route 49.0.0.0 255.0.0.0 Null0
ip route 50.0.0.0 255.0.0.0 Null0
ip route 58.0.0.0 255.0.0.0 Null0
ip route 59.0.0.0 255.0.0.0 Null0
```

```
ip route 70.0.0.0 255.0.0.0 Null0
ip route 71.0.0.0 255.0.0.0 Null0
ip route 72.0.0.0 255.0.0.0 Null0
ip route 73.0.0.0 255.0.0.0 Null0
ip route 74.0.0.0 255.0.0.0 Null0
ip route 75.0.0.0 255.0.0.0 Null0
ip route 76.0.0.0 255.0.0.0 Null0
ip route 77.0.0.0 255.0.0.0 Null0
ip route 78.0.0.0 255.0.0.0 Null0
ip route 79.0.0.0 255.0.0.0 Null0
ip route 83.0.0.0 255.0.0.0 Null0
ip route 84.0.0.0 255.0.0.0 Null0
ip route 85.0.0.0 255.0.0.0 Null0
ip route 86.0.0.0 255.0.0.0 Null0
ip route 87.0.0.0 255.0.0.0 Null0
ip route 88.0.0.0 255.0.0.0 Null0
ip route 89.0.0.0 255.0.0.0 Null0
ip route 90.0.0.0 255.0.0.0 Null0
ip route 91.0.0.0 255.0.0.0 Null0
ip route 92.0.0.0 255.0.0.0 Null0
ip route 93.0.0.0 255.0.0.0 Null0
ip route 94.0.0.0 255.0.0.0 Null0
ip route 95.0.0.0 255.0.0.0 Null0
ip route 96.0.0.0 255.0.0.0 Null0
ip route 97.0.0.0 255.0.0.0 Null0
ip route 98.0.0.0 255.0.0.0 Null0
ip route 99.0.0.0 255.0.0.0 Null0
ip route 100.0.0.0 255.0.0.0 Null0
ip route 101.0.0.0 255.0.0.0 Null0
ip route 102.0.0.0 255.0.0.0 Null0
ip route 103.0.0.0 255.0.0.0 Null0
ip route 104.0.0.0 255.0.0.0 Null0
ip route 105.0.0.0 255.0.0.0 Null0
ip route 106.0.0.0 255.0.0.0 Null0
ip route 107.0.0.0 255.0.0.0 Null0
ip route 108.0.0.0 255.0.0.0 Null0
ip route 109.0.0.0 255.0.0.0 Null0
ip route 110.0.0.0 255.0.0.0 Null0
ip route 111.0.0.0 255.0.0.0 Null0
ip route 112.0.0.0 255.0.0.0 Null0
ip route 113.0.0.0 255.0.0.0 Null0
ip route 114.0.0.0 255.0.0.0 Null0
ip route 115.0.0.0 255.0.0.0 Null0
ip route 116.0.0.0 255.0.0.0 Null0
ip route 117.0.0.0 255.0.0.0 Null0
ip route 118.0.0.0 255.0.0.0 Null0
ip route 119.0.0.0 255.0.0.0 Null0
ip route 120.0.0.0 255.0.0.0 Null0
ip route 121.0.0.0 255.0.0.0 Null0
ip route 122.0.0.0 255.0.0.0 Null0
ip route 123.0.0.0 255.0.0.0 Null0
ip route 124.0.0.0 255.0.0.0 Null0
ip route 125.0.0.0 255.0.0.0 Null0
ip route 126.0.0.0 255.0.0.0 Null0
ip route 127.0.0.0 255.0.0.0 Null0
ip route 169.254.0.0 255.255.0.0 Null0
ip route 172.0.0.0 255.0.0.0 Null0
```

```
ip route 172.16.0.0 255.240.0.0 Null0
ip route 173.0.0.0 255.0.0.0 Null0
ip route 174.0.0.0 255.0.0.0 Null0
ip route 175.0.0.0 255.0.0.0 Null0
ip route 176.0.0.0 255.0.0.0 Null0
ip route 177.0.0.0 255.0.0.0 Null0
ip route 178.0.0.0 255.0.0.0 Null0
ip route 179.0.0.0 255.0.0.0 Null0
ip route 180.0.0.0 255.0.0.0 Null0
ip route 181.0.0.0 255.0.0.0 Null0
ip route 182.0.0.0 255.0.0.0 Null0
ip route 183.0.0.0 255.0.0.0 Null0
ip route 184.0.0.0 255.0.0.0 Null0
ip route 185.0.0.0 255.0.0.0 Null0
ip route 186.0.0.0 255.0.0.0 Null0
ip route 187.0.0.0 255.0.0.0 Null0
ip route 188.0.0.0 255.0.0.0 Null0
ip route 189.0.0.0 255.0.0.0 Null0
ip route 190.0.0.0 255.0.0.0 Null0
ip route 192.0.2.0 255.255.255.0 Null0
ip route 192.168.0.0 255.255.0.0 Null0
ip route 197.0.0.0 255.0.0.0 Null0
ip route 223.0.0.0 255.0.0.0 Null0
no ip http server
no ip http secure-server
!
no ip access-list extended antinoise
ip access-list extended antinoise
 remark Anti-Noise ACL
 remark Knock down noisy and undesired traffic at the earliest opportunity
 remark NetBIOS provides the largest amount of noise
 deny   tcp any any range 137 139 log-input
 deny   tcp any any eq 135 log-input
 deny   udp any any eq 135 log-input
 deny   tcp any any eq 445 log-input
 deny   udp any any eq 445 log-input
 remark MS-SQL also has a lot of traffic
 deny   tcp any any eq 1433 log-input
 deny   udp any any eq 1434 log-input
 remark TCP and UDP small services
 deny   tcp any any range 1 19 log-input
 deny   udp any any range 1 19 log-input
 remark DHCP
 deny   tcp any any range 67 68 log-input
 deny   udp any any range bootps bootpc log-input
 remark SNMP - must permit from GIACE addresses
 permit tcp any 1.2.2.0 0.0.1.255 range 161 162 log-input
 permit udp any 1.2.2.0 0.0.1.255 range snmp snmptrap log-input
 deny   tcp any any range 161 162 log-input
 deny   udp any any range snmp snmptrap log-input
 remark TFTP is often used as a backchannel following a breach
 deny   udp any any eq tftp log-input
 remark IRC is often used as a worm backchannel/control channel
 deny   tcp any any eq irc log-input
 deny   udp any any eq 194 log-input
 deny   tcp any any eq 994 log-input
 deny   udp any any eq 994 log-input
```

```
 deny   tcp any any eq 6667 log-input
 deny   udp any any eq 6667 log-input
 deny   tcp any any eq 7666 log-input
 remark Block telnet for added safety
 deny   tcp any any eq telnet log-input
 remark Let everything else through
 permit ip any any
no ip access-list extended antispoof-egress
ip access-list extended antispoof-egress
 remark Anti-Spoofing - egress traffic
 remark Allow only traffic coming from an address assigned to GIACE
 permit ip 1.2.2.0 0.0.1.255 any
 deny   ip any any log-input
no ip access-list extended antispoof-ingress
ip access-list extended antispoof-ingress
 remark Anti-Spoof (Incoming)
 remark Knock down RFC-1918, unassigned, and otherwise undesirables on the
      way into the network
 remark Blocks are aggregated to help speed ACL processing.
 permit ip 1.2.2.0 0.0.1.255 any
 deny   ip 1.0.0.0 0.255.255.255 any log-input
 deny   ip 2.0.0.0 0.255.255.255 any log-input
 deny   ip 5.0.0.0 0.255.255.255 any log-input
 deny   ip 7.0.0.0 0.255.255.255 any log-input
 deny   ip 10.0.0.0 0.255.255.255 any log-input
 deny   ip 23.0.0.0 0.255.255.255 any log-input
 deny   ip 27.0.0.0 0.255.255.255 any log-input
 deny   ip 31.0.0.0 0.255.255.255 any log-input
 deny   ip 36.0.0.0 1.255.255.255 any log-input
 deny   ip 39.0.0.0 0.255.255.255 any log-input
 deny   ip 41.0.0.0 0.255.255.255 any log-input
 deny   ip 42.0.0.0 0.255.255.255 any log-input
 deny   ip 49.0.0.0 0.255.255.255 any log-input
 deny   ip 50.0.0.0 0.255.255.255 any log-input
 deny   ip 58.0.0.0 1.255.255.255 any log-input
 deny   ip 70.0.0.0 1.255.255.255 any log-input
 deny   ip 72.0.0.0 7.255.255.255 any log-input
 deny   ip 83.0.0.0 0.255.255.255 any log-input
 deny   ip 84.0.0.0 3.255.255.255 any log-input
 deny   ip 88.0.0.0 7.255.255.255 any log-input
 deny   ip 96.0.0.0 31.255.255.255 any log-input
 deny   ip 169.254.0.0 0.0.255.255 any log-input
 deny   ip 172.16.0.0 0.15.255.255 any log-input
 deny   ip 173.0.0.0 0.255.255.255 any log-input
 deny   ip 174.0.0.0 1.255.255.255 any log-input
 deny   ip 176.0.0.0 7.255.255.255 any log-input
 deny   ip 184.0.0.0 3.255.255.255 any log-input
 deny   ip 189.0.0.0 0.255.255.255 any log-input
 deny   ip 190.0.0.0 0.255.255.255 any log-input
 deny   ip 192.0.2.0 0.0.0.255 any log-input
 deny   ip 192.168.0.0 0.0.255.255 any log-input
 deny   ip 197.0.0.0 0.255.255.255 any log-input
 deny   ip 223.0.0.0 0.255.255.255 any log-input
 deny   ip 224.0.0.0 31.255.255.255 any log-input
 remark Kill ICMP fragments - should never be necessary to frag ICMP
 deny   icmp any any log-input fragments
 remark GIACE's multicast usage is local only
```

```
 deny   ip any 224.0.0.0 15.255.255.255
 remark Let everything else through, at least to the firewall
 permit ip any any log-input
! send copious logs to syslog
logging trap debugging
logging facility local5
! firewall will nat logs to the logging server
logging host 1.2.3.2
no access-list 20
access-list 20 remark SNMP controls
access-list 20 permit 1.2.3.2
access-list 20 permit 1.2.3.251
access-list 20 permit 1.2.3.252
access-list 20 deny   any log
no access-list 30
access-list 30 remark Multicast boundary definition
access-list 30 deny   224.0.0.0 0.0.16.255
! help prevent traffic based DoS attacks as best possible
no access-list 150
access-list 150 remark Rate-Limit UDP
access-list 150 permit udp any any
no access-list 160
access-list 160 remark Rate-Limit ICMP
access-list 160 permit icmp any any
! Squash Cisco's pressing need to advertise system information
no cdp run
! Configure SNMP and activate traps
snmp-server engineID local 0000000902000002166 47FB5
snmp-server community GIACE-strong-snmp RO 20
snmp-server enable traps snmp
snmp-server enable traps hsrp
snmp-server enable traps config
snmp-server enable traps syslog
! server to send SNMP traps to - again, firewall will NAT in...
snmp-server host 1.2.3.2 bdr-routers
!
banner motd #

Router GIACE-bdr-1.  Access to this device or attached networks by
unauthorized users is prohibited.  No user of this device or connected
networks has any expectation of privacy.
Violators will be prosecuted.

#
!
line con 0
 transport input none
line aux 0
 transport input none
 transport output none
line vty 0 4
 no login
 transport input none
 transport output none
!
! setup ntp in authentication mode - go in to DMZ servers
ntp authentication-key 1 md5 09107D2C3A3732263427211375 7
```

```
ntp authenticate
ntp server 1.2.3.22
ntp server 1.2.3.21 prefer
end
```

Appendix C: Firewall Configuration

```
interface ethernet0 100full
interface ethernet1 100full
interface ethernet2 100full
interface ethernet3 100full
nameif ethernet0 outside security0
nameif ethernet1 dmz security20
nameif ethernet2 inside security100
nameif ethernet3 failover security10
enable password <password> encrypted
 password <password> encrypted
hostname GIACE-fw-a
domain-name giace.com
clock timezone EST -5
fixup protocol dns maximum-length 512
fixup protocol ftp
fixup protocol http 80
no fixup protocol h323 h225
no fixup protocol h323 ras
no fixup protocol ils
no fixup protocol rsh
fixup protocol rtsp 554
fixup protocol smtp 25
no fixup protocol sqlnet 1521
no fixup protocol sip 5060
no fixup protocol skinny 2000
no access-list inbound compiled
access-list inbound permit tcp any host 1.2.3.10 eq www
access-list inbound permit tcp any host 1.2.3.11 eq 443
access-list inbound permit tcp any host 1.2.3.20 eq smtp
access-list inbound permit tcp any host 1.2.3.21 eq smtp
access-list inbound permit tcp any host 1.2.3.50 eq 443
access-list inbound permit 51 1.2.3.252 255.255.255.254 host 1.2.3.241
access-list inbound permit 51 1.2.3.252 255.255.255.254 host 1.2.3.244
access-list inbound permit 51 host 1.2.3.240 host 1.2.3.242
no access-list ipsec-remote compiled
access-list ipsec-remote permit ip 192.168.0.0 255.255.0.0 192.168.253.0
      255.255.255.192
no access-list ipsec-internal compiled
access-list ipsec-internal permit ip 192.168.129.252 255.255.255.254
      192.168.193.0 255.255.255.0
access-list outbound-dmz compiled
access-list outbound-dmz deny tcp any any eq 6667 log default
access-list outbound-dmz deny udp any any eq 67 log default
access-list outbound-dmz permit tcp host 192.168.65.20 host 192.168.129.30 eq
      25
access-list outbound-dmz permit tcp host 192.168.65.21 host 192.168.129.30 eq
      25
access-list outbound-dmz permit tcp host 192.168.65.40 host 192.168.1.20 eq
      1433
access-list outbound-dmz permit tcp host 192.168.65.41 host 192.168.1.20 eq
      1433
```

```
access-list outbound-dmz permit 51 192.168.65.0 255.255.255.0 host
      192.168.193.11
access-list outbound-dmz permit 51 192.168.65.0 255.255.255.0 host
      192.168.193.14
access-list outbound-dmz permit 51 host 192.168.65.240 host 192.168.193.12
access-list outbound-dmz deny ip any 192.168.0.0 255.255.0.0 log
access-list outbound-dmz permit udp host 192.168.65.20 any eq 53
access-list outbound-dmz permit udp host 192.168.65.21 any eq 53
access-list outbound-dmz permit udp host 192.168.65.20 any eq 123
access-list outbound-dmz permit udp host 192.168.65.21 any eq 123
access-list outbound-dmz permit tcp host 192.168.65.20 any eq 80
access-list outbound-dmz permit tcp host 192.168.65.21 any eq 80
access-list outbound-dmz permit tcp host 192.168.65.20 any eq 443
access-list outbound-dmz permit tcp host 192.168.65.21 any eq 443
access-list outbound-dmz permit tcp host 192.168.65.20 any eq 25
access-list outbound-dmz permit tcp host 192.168.65.21 any eq 25
access-list outbound-internal compiled
access-list outbound-internal deny tcp any any eq 6667 log default
access-list outbound-internal deny udp any any eq 67 log default
access-list outbound-internal permit udp host 192.168.129.20 any eq 53
access-list outbound-internal permit udp host 192.168.129.30 any eq 53
access-list outbound-internal permit udp host 192.168.129.20 any eq 123
access-list outbound-internal permit udp host 192.168.129.30 any eq 123
access-list outbound-internal permit tcp host 192.168.193.11 any eq 25
access-list outbound-internal permit tcp host 192.168.193.12 any eq 25
access-list outbound-internal permit tcp 192.168.0.0 255.255.0.0 host
      192.168.65.20 eq 3128
access-list outbound-internal permit tcp 192.168.0.0 255.255.0.0 host
      192.168.65.21 eq 3128
access-list outbound-internal permit 51 host 192.168.193.11 192.168.65.0
      255.255.255.0
access-list outbound-internal permit 51 host 192.168.193.14 192.168.65.0
      255.255.255.0
access-list outbound-internal permit 51 host 192.168.193.11 1.2.3.252
      255.255.255.254
access-list outbound-internal permit 51 host 192.168.193.14 1.2.3.252
      255.255.255.254
access-list outbound-internal permit 51 host 192.168.193.12 host 1.2.3.240
access-list outbound-internal permit tcp any 192.168.65.0 255.255.255.0 eq 22
access-list outbound-internal permit tcp any 192.168.65.0 255.255.255.0 eq 22
access-list outbound-internal permit tcp any 192.168.65.0 255.255.255.0 eq 80
access-list outbound-internal permit tcp any 192.168.65.0 255.255.255.0 eq 80
access-list outbound-internal permit tcp any 192.168.65.0 255.255.255.0 eq
      443
access-list outbound-internal permit tcp any 192.168.65.0 255.255.255.0 eq
      443
access-group inbound in interface external
access-group outbound-dmz in interface dmz
access-group outbound-internal in interface internal
pager lines 24
logging on
logging facility 19
logging buffered 4
logging console 5
logging host inside 192.168.193.14
logging trap 5
logging standby
```

```
mtu outside 1500
mtu inside 1500
mtu dmz 1500
mtu failover 1500
ip address outside 1.2.3.250 255.255.254.0
ip address inside 192.168.129.252 255.255.254.0
ip address dmz 192.168.65.252 255.255.255.0
ip address failover 192.168.254.1 255.255.255.0
ip local pool remote-vpn 192.168.253.10-192.168.253.59
ip audit info action alarm
ip audit attack action reset
fragment chain 1 outside
fragment chain 1 inside
fragment chain 1 dmz
failover
failover ip address outside 1.2.3.251
failover ip address inside 192.168.129.253
failover ip address dmz 192.168.65.253
failover ip address failover 192.168.254.2
failover mac address outside <act_mac> <standby_mac>
failover mac address inside <act_mac> <standby_mac>
failover mac address dmz <act_mac> <standby_mac>
failover link failover
failover poll 5
failover lan interface failover
failover lan unit primary
failover replicate http
arp timeout 14400
global (outside) 65  1.2.3.70-79   netmask 255.255.254.0
global (outside) 193 1.2.3.180-189 netmask 255.255.254.0
nat (inside) 193 192.168.193.0 255.255.255.0 0 0
nat (dmz) 65 192.168.65.0 19 0 0
nat (inside) 0 access-list ipsec-remote
static (dmz, outside) tcp 1.2.3.10 www 192.168.65.10 www netmask
      255.255.255.255
static (dmz, outside) tcp 1.2.3.11 443 192.168.65.11 443 netmask
      255.255.255.255
static (dmz, outside) tcp 1.2.3.20 smtp 192.168.65.20 smtp netmask
      255.255.255.255
static (dmz, outside) tcp 1.2.3.21 smtp 192.168.65.21 smtp netmask
      255.255.255.255
static (dmz, outside) tcp 1.2.3.50 443 192.168.65.50 443 netmask
      255.255.255.255
static (inside, outside) 1.2.3.241 192.168.193.11 netmask 255.255.255.255
static (inside, outside) 1.2.3.242 192.168.193.12 netmask 255.255.255.255
static (inside, outside) 1.2.3.244 192.168.193.14 netmask 255.255.255.255
route outside 0.0.0.0 0.0.0.0 1.2.3.1 1
timeout xlate 3:00:00
timeout conn 1:00:00 half-closed 0:10:00 udp 0:02:00 rpc 0:10:00 h225 1:00:00
timeout h323 0:05:00 mgcp 0:05:00 sip 0:30:00 sip_media 0:02:00
timeout uauth 0:05:00 absolute
aaa-server authenex protocol radius
aaa-server authenex host 192.168.193.15 <radius_key>
aaa-server radius-acctport 1813           :: ports used by authenex radius
aaa-server radius-authport 1812
no http server enable
no snmp-server community public
```

```
no snmp-server community private
no snmp-server community cisco
no snmp-server community write
snmp-server location GIACE
snmp-server contact Security Administrator
snmp-server community <giace_strong_community>
snmp-server host 192.168.193.11 trap
snmp-server host 192.168.193.11
snmp-server enable traps
floodguard enable
sysopt permit-ipsec
sysopt security fragguard
crypto ipsec transform-set internal ah-md5-hmac mode transport
crypto ipsec transform-set remote esp-aes esp-md5-hmac
crypto map internal 10 ipsec-isakmp
crypto map internal 10 match address ipsec-internal
crypto map internal 10 set transform-set internal
crypto map internal 10 set peer 192.168.193.0
crypto map internal 10 interface internal
crypto map remote 20 ipsec-isakmp dynamic dyn-remote
crypto map remote client token authentication authenex
crypto map remote client configuration address initiate
crypto map remote 20 interface inside
crypto dynamic-map dyn-remote 20 match address ipsec-remote
crypto dynamic-map dyn-remote 20 set transform-set internal
crypto dynamic-map dyn-remote 20 set pfs group2
isakmp enable outside
isakmp enable inside
isakmp policy 10 encryption aes
isakmp policy 10 group 2
isakmp key <strong_key1> address 0.0.0.0 netmask 0.0.0.0
isakmp key <strong_key2> address 192.168.193.0 netmask 255.255.255.0
vpngroup remote-ipsec address-pool remote-vpn
vpngroup remote-ipsec dns-server 192.168.129.20
vpngroup remote-ipsec dns-server 192.168.129.30
vpngroup remote-ipsec default-domain giace.com
vpngroup remote-ipsec split-tunnel ipsec-remote
vpngroup remote-ipsec idle-time 1800
vpngroup remote-ipsec max-time 43200
no telnet
no ssh
```

Appendix D: Internal Router Configuration

The following is a sample configuration for a MSFC for the Catalyst 6500 core switch designed to perform as a filtering router in the core. Filtering at the core is not Cisco's recommended configuration, however the relatively small nature of the GIACE network compared to Cisco's case studies justifies collapsing the access, distribution, and core layers into a single device.

```
! Anti-congestion algoritm
service nagle
no service pad
! Verbose logging timestamps
service timestamps debug datetime msec localtime show-timezone
service timestamps log datetime msec localtime show-timezone
! Encrypt all passwords
service password-encryption
no service dhcp
!
hostname GIACE-core-1
!
logging buffered 16384 notifications
! Crackers get a life!  Stronger passwords will be used in the final config.
enable secret 5 $1$lTyH$GCuhuZK3agVQtfGZ5maJj1
!
username GIACEuser password 7 104D000A0618
!
!
!
!
clock timezone EST -5
ip subnet-zero
no ip source-route
no ip finger
no ip domain-lookup
!
no ip bootp server
!
interface Null 0
 no ip unreachables
!
interface vlan1
 description Database VLAN Interface
 ip address 192.168.1.253 255.255.255.0
 ip access-group antinoise in
 ip access-group to-db out
 no ip proxy-arp
 ip accounting access-violations
 ip route-cache flow
 no cdp enable
! enable HSRP for failover
 standby authentication database
 standby name GIACE-db-gw
```

```
 standby ip 192.168.1.1
!
interface vlan129
 description Internal VLAN Interface
 ip address 192.168.129.253 255.255.254.0
 ip access-group antinoise in
 no ip proxy-arp
 ip accounting access-violations
 ip route-cache flow
 no cdp enable
! enable HSRP for failover
 standby authentication internal
 standby name GIACE-int-gw
 standby ip 192.168.129.1
!
interface vlan193
 description Administrative VLAN Interface
 ip address 192.168.193.253 255.255.255.0
 ip access-group antinoise in
 ip access-group to-admin out
 no ip proxy-arp
 ip accounting access-violations
 ip route-cache flow
 no cdp enable
! enable HSRP for failover
 standby authentication admin
 standby name GIACE-admin-gw
 standby ip 192.168.193.1
!
ip classless
ip route 0.0.0.0 0.0.0.0 192.168.129.1
no ip finger
no ip domain-lookup
no ip http server
no ip http secure-server
!
no ip access-list extended to-db
ip access-list extended to-db
 remark This access list defines traffic allowed to enter the DB vlan
 remark Apply to VLAN1 out
 permit tcp any any established
 remark --- Allow database access from all internal hosts
 permit tcp 192.168.0.0 0.0.255.255 any eq 1433
 permit udp 192.168.0.0 0.0.255.255 any eq 1434
 remark --- Allow NTP time sync
 permit udp host 192.168.129.21 any eq 123
 permit udp host 192.168.129.31 any eq 123
 remark --- Allow DNS query responses
 permit udp host 192.168.129.21 any eq 53
 permit udp host 192.168.129.31 any eq 53
 remark --- Allow a host of ports for AD
 permit udp host 192.168.129.21 any eq 135
 permit udp host 192.168.129.21 any eq 88
 permit udp host 192.168.129.21 any eq 389
 remark --- Backup software should connect over above defined SQL ports
 remark --- If more access necessary, add here
 deny   ip any any log-input
```

```
!
no ip access-list extended to-admin
ip access-list extended to-admin
 remark This access list defined traffic allowed to enter the ADMIN vlan
 remark Apply to VLAN193 out - not yet performance tuned!
 remark Hosts on the ADMIN network are implied permit, including the MSFCs
 remark This list only contains currently defined devices - will be more!
 permit tcp any any established
 remark --- Allow syslog only from particular devices - don't want the
      possiblity of a DoS!
 permit udp 1.2.3.252 0.0.0.1 host 192.168.193.14 eq 514
 permit udp 192.168.1.20 0.0.0.1 host 192.168.193.14 eq 514
 permit udp 192.168.1.40 0.0.0.1 host 192.168.193.14 eq 514
 permit udp 192.168.65.20 0.0.0.1 host 192.168.193.14 eq 514
 permit udp 192.168.65.30 0.0.0.1 host 192.168.193.14 eq 514
 permit udp 192.168.65.40 0.0.0.1 host 192.168.193.14 eq 514
 permit udp host 192.168.65.50 host 192.168.193.14 eq 514
 permit udp host 192.168.65.60 host 192.168.193.14 eq 514
 permit udp 192.168.65.250 0.0.0.1 host 192.168.193.14 eq 514
 permit udp 192.168.65.252 0.0.0.1 host 192.168.193.14 eq 514
 permit udp host 192.168.129.10 host 192.168.193.14 eq 514
 permit udp host 192.168.129.20 host 192.168.193.14 eq 514
 permit udp host 192.168.129.30 host 192.168.193.14 eq 514
 permit udp 192.168.129.252 0.0.0.1 host 192.168.193.14 eq 514
 remark --- Allow time/DNS responses from appropriate servers
 permit udp host 192.168.129.20 any eq 53
 permit udp host 192.168.129.30 any eq 53
 permit udp host 192.168.129.20 any eq 123
 permit udp host 192.168.129.30 any eq 123
 remark --- Allow a host of ports for AD
 permit udp host 192.168.129.21 any eq 135
 permit udp host 192.168.129.21 any eq 88
 permit udp host 192.168.129.21 any eq 389
 remark --- Allow SNMP information to monitor station
 permit udp 1.2.3.252 0.0.0.1 host 192.168.193.11 range 161 162
 permit udp 192.168.1.20 0.0.0.1 host 192.168.193.11 range 161 162
 permit udp 192.168.1.40 0.0.0.1 host 192.168.193.11 range 161 162
 permit udp 192.168.65.20 0.0.0.1 host 192.168.193.11 range 161 162
 permit udp 192.168.65.30 0.0.0.1 host 192.168.193.11 range 161 162
 permit udp 192.168.65.40 0.0.0.1 host 192.168.193.11 range 161 162
 permit udp host 192.168.65.50 host 192.168.193.11 range 161 162
 permit udp host 192.168.65.60 host 192.168.193.11 range 161 162
 permit udp 192.168.65.250 0.0.0.1 host 192.168.193.11 range 161 162
 permit udp 192.168.65.252 0.0.0.1 host 192.168.193.11 range 161 162
 permit udp host 192.168.129.10 host 192.168.193.11 range 161 162
 permit udp host 192.168.129.20 host 192.168.193.11 range 161 162
 permit udp host 192.168.129.30 host 192.168.193.11 range 161 162
 permit udp 192.168.129.252 0.0.0.1 host 192.168.193.11 range 161 162
 remark --- Allow RADIUS from firewalls to auth server
 permit tcp 192.168.129.252 0.0.0.1 host 192.168.193.15 range 1812 1813
 permit udp 192.168.129.252 0.0.0.1 host 192.168.193.15 range 1812 1813
 remark --- Allow IDS sensor logging to IDS console - runs on mysql
 permit tcp host 1.2.3.240 host 192.168.193.12 eq 3306
 permit tcp host 192.168.65.240 host 192.168.193.12 eq 3306
 remark --- backups should be all outbound connections, if not add rules here
 deny   ip any any log-input
!
```

```
no ip access-list extended antinoise
ip access-list extended antinoise
 remark Knock down traffic that we never want to cross the core router
 remark Apply to in on all interfaces
 remark --- no use for tcp/udp small services
 deny   tcp any any range 1 19 log-input
 deny   udp any any range 1 19 log-input
 remark --- TFTP has some uses - may want to enable on a case-by-case
 deny   udp any any eq 69 log-input
 remark --- IRC Bad - control channel for worms
 deny   tcp any any eq 194 log-input
 deny udp any any eq 194 log-input
 deny tcp any any eq 994 log-input
 deny udp any any eq 994 log-input
 deny tcp any any eq 6667 log-input
 deny udp any any eq 6667 log-input
 deny tcp any any eq 7776 log-input
 remark --- let the rest at least get to the outbound ACL
 permit ip any any
!
! send copious logs to syslog
logging trap debugging
logging facility local5
logging host 192.168.193.14
!
no access-list 20
access-list 20 remark SNMP controls
access-list 20 permit 192.168.193.11
access-list 20 deny   any log
!
! Squash Cisco's pressing need to advertise system information
no cdp run
! Configure SNMP and activate traps
snmp-server engineID local 00000009020000216647FB5
snmp-server community GIACE-strong-snmp RO 20
snmp-server enable traps snmp
snmp-server enable traps hsrp
snmp-server enable traps config
snmp-server enable traps syslog
snmp-server host 192.168.193.11 core-routers
!
banner motd #

Router GIACE-core-1.  Access to this device or attached networks by
unauthorized users is prohibited.  No user of this device or connected
networks has any expectation of privacy.
Violators will be prosecuted.

#
!
line con 0
 transport input none
line aux 0
 transport input none
 transport output none
line vty 0 4
 no login
```

```
 transport input none
 transport output none
!
! setup ntp in authentication mode
ntp authentication-key 1 md5 09107D2C3A3732263427211375 7
ntp authenticate
ntp server 192.168.129.31
ntp server 192.168.129.21 prefer
end
```

Appendix E: IPTABLES Samples

IPTABLES Load Script for www-a

As mentioned above, this script is monolithic in nature. GIACE administrators may wish to break into a hierarchical set of scripts to help facilitate DMZ-wide changes to the local firewall rules.

```
# Machine Specific Configuration

LOC_HOSTNAME="www-a"
LOC_IP="192.168.65.31"

# Policy to deny for inbound and forward:
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT ACCEPT

#--------------------------------------------------------------------

# VALID table contains the only rules which should be changed from machine
# to machine.  All others tables are standard within the DMZ machines.
# VALID is also the only table which should contain accept rules.
# Create the VALID table
iptables -N VALID

# Heartbeat for HA clustering.  HA runs multicast, so this rule can be
# further limited to a single address for each cluster once HA is configured.
iptables -A VALID -p udp -m udp -d $LOC_IP --dport 694 -j ACCEPT

# Obviously need HTTP allowed
iptables -A VALID -p tcp -m tcp -d $LOC_IP --dport 80 -j ACCEPT

# Time/DNS are UDP and therefore not stateful, so we must
# explicitly accept responses:
iptables -A VALID -p udp -m udp -s 192.168.65.21 -d $LOC_IP --dport 123 -j
      ACCEPT
iptables -A VALID -p udp -m udp -s 192.168.65.22 -d $LOC_IP --dport 123 -j
      ACCEPT
iptables -A VALID -p udp -m udp -s 192.168.65.21 -d $LOC_IP --dport 53 -j
      ACCEPT
iptables -A VALID -p udp -m udp -s 192.168.65.22 -d $LOC_IP --dport 53 -j
      ACCEPT

# IPSec for logging and other admin stuff
iptables -A VALID -p 50 -s 192.168.193.0/24 -d $LOC_IP -j ACCEPT
iptables -A VALID -p 51 -s 192.168.193.0/24 -d $LOC_IP -j ACCEPT
iptables -A VALID -p udp -m udp -s 192.168.193.0/24 --sport 500 -d $LOC_IP --
      dport 500 -j ACCEPT

# SSH for management - only from monitor workstation.
# Might add select administrator workstations later.
```

```
iptables -A VALID -p tcp -m tcp -s 192.168.193.11 -d $LOC_IP --dport 22 -j
      ACCEPT

#-------------------------------------------------------------------------

# Create the LOGDROP table
iptables -N LOGDROP

# Log every packet that reaches this table verbosely,
# prefixing with the system hostname.
iptables -A LOGDROP -j LOG --log-prefix $LOC_HOSTNAME --log-tcp-options --
      log-ip-options

# Drop every packet that reaches this table.
iptables -A LOGDROP -j DROP

#-------------------------------------------------------------------------

# Create the BAN table
iptables -N BAN

# A couple noisy ISP blocks - SAVECOM-NET, UNICOM
iptables -A BAN -s 61.65.0.0/255.255.128.0 -j LOGDROP
iptables -A BAN -s 61.240.0.0/255.252.0.0 -j LOGDROP
# Return if no match

#-------------------------------------------------------------------------

# Create the INVAL_SRC table
iptables -N INVAL_SRC

# Short circuit known good addresses for better performance.
# We may elect to filter more of these with later rules.
# These rules are order independent among each other,
# but all should remain at the top for performance.
iptables -A INVAL_SRC -s 192.168.1.0/255.255.255.0 -j RETURN
iptables -A INVAL_SRC -s 192.168.65.0/255.255.255.0 -j RETURN
iptables -A INVAL_SRC -s 192.168.129.0/255.255.254.0 -j RETURN
iptables -A INVAL_SRC -s 192.168.193.0/255.255.255.0 -j RETURN

# Start stripping illegal and unassigned addresses.
# This is not the place to block naughty but valid addresses -
# the BAN table exists for that.
# The following rules may be reordered at will if necessary.
iptables -A INVAL_SRC -s 1.0.0.0/255.0.0.0 -j LOGDROP
iptables -A INVAL_SRC -s 2.0.0.0/255.0.0.0 -j LOGDROP
iptables -A INVAL_SRC -s 5.0.0.0/255.0.0.0 -j LOGDROP
iptables -A INVAL_SRC -s 7.0.0.0/255.0.0.0 -j LOGDROP
iptables -A INVAL_SRC -s 10.0.0.0/255.0.0.0 -j LOGDROP
iptables -A INVAL_SRC -s 23.0.0.0/255.0.0.0 -j LOGDROP
iptables -A INVAL_SRC -s 27.0.0.0/255.0.0.0 -j LOGDROP
iptables -A INVAL_SRC -s 31.0.0.0/255.0.0.0 -j LOGDROP
iptables -A INVAL_SRC -s 36.0.0.0/254.0.0.0 -j LOGDROP
iptables -A INVAL_SRC -s 39.0.0.0/255.0.0.0 -j LOGDROP
iptables -A INVAL_SRC -s 41.0.0.0/255.0.0.0 -j LOGDROP
iptables -A INVAL_SRC -s 42.0.0.0/255.0.0.0 -j LOGDROP
iptables -A INVAL_SRC -s 49.0.0.0/255.0.0.0 -j LOGDROP
```

```
iptables -A INVAL_SRC -s 50.0.0.0/255.0.0.0 -j LOGDROP
iptables -A INVAL_SRC -s 58.0.0.0/254.0.0.0 -j LOGDROP
iptables -A INVAL_SRC -s 70.0.0.0/254.0.0.0 -j LOGDROP
iptables -A INVAL_SRC -s 72.0.0.0/248.0.0.0 -j LOGDROP
iptables -A INVAL_SRC -s 83.0.0.0/255.0.0.0 -j LOGDROP
iptables -A INVAL_SRC -s 84.0.0.0/252.0.0.0 -j LOGDROP
iptables -A INVAL_SRC -s 88.0.0.0/248.0.0.0 -j LOGDROP
iptables -A INVAL_SRC -s 96.0.0.0/254.0.0.0 -j LOGDROP
iptables -A INVAL_SRC -s 169.254.0.0/255.255.0.0 -j LOGDROP
iptables -A INVAL_SRC -s 172.16.0.0/255.240.0.0 -j LOGDROP
iptables -A INVAL_SRC -s 173.0.0.0/255.0.0.0 -j LOGDROP
iptables -A INVAL_SRC -s 174.0.0.0/254.0.0.0 -j LOGDROP
iptables -A INVAL_SRC -s 176.0.0.0/248.0.0.0 -j LOGDROP
iptables -A INVAL_SRC -s 184.0.0.0/252.0.0.0 -j LOGDROP
iptables -A INVAL_SRC -s 189.0.0.0/255.0.0.0 -j LOGDROP
iptables -A INVAL_SRC -s 190.0.0.0/255.0.0.0 -j LOGDROP
iptables -A INVAL_SRC -s 192.0.2.0/255.255.255.0 -j LOGDROP
iptables -A INVAL_SRC -s 192.168.0.0/255.255.0.0 -j LOGDROP
iptables -A INVAL_SRC -s 197.0.0.0/255.0.0.0 -j LOGDROP
iptables -A INVAL_SRC -s 223.0.0.0/255.0.0.0 -j LOGDROP
# Return to calling rule for more processing.

#------------------------------------------------------------------------

# Main INPUT table
# Look for banned sites in the BAN table first.
# There is a reason they are banned!
iptables -A INPUT -j BAN

# Next strip RFC-1918, other invalid, or unassigned IP source ranges
# as defined in the INVAL_SRC table.
iptables -A INPUT -j INVAL_SRC

# Trash packets for connections not properly registered in the state table.
# (I.E. SYN-FIN, etc.)
iptables -A INPUT -m state --state INVALID -j LOGDROP

# Disallow fragments - the routers should have reassembled already
iptables -A INPUT -f -j LOGDROP

# Look for allowable traffic on the VALID table
iptables -A INPUT -j VALID

# Log and drop everything else
iptables -A INPUT -j LOGDROP
```

Loaded Tables

The following table display is the result of running the above script:

```
#iptables -L -n -v
Chain INPUT (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 BAN        all  --  *      *       0.0.0.0/0            0.0.0.0/0
    0     0 INVAL_SRC  all  --  *      *       0.0.0.0/0            0.0.0.0/0
    0     0 LOGDROP    all  --  *      *       0.0.0.0/0            0.0.0.0/0          state INVALID
    0     0 LOGDROP    all  -f  *      *       0.0.0.0/0            0.0.0.0/0
    0     0 VALID      all  --  *      *       0.0.0.0/0            0.0.0.0/0
    0     0 LOGDROP    all  --  *      *       0.0.0.0/0            0.0.0.0/0

Chain FORWARD (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain BAN (1 references)
 pkts bytes target     prot opt in     out     source               destination
    0     0 LOGDROP    all  --  *      *       61.65.0.0/17         0.0.0.0/0
    0     0 LOGDROP    all  --  *      *       61.240.0.0/14        0.0.0.0/0

Chain INVAL_SRC (1 references)
 pkts bytes target     prot opt in     out     source               destination
    0     0 RETURN     all  --  *      *       192.168.1.0/24       0.0.0.0/0
    0     0 RETURN     all  --  *      *       192.168.65.0/24      0.0.0.0/0
    0     0 RETURN     all  --  *      *       192.168.128.0/23     0.0.0.0/0
    0     0 RETURN     all  --  *      *       192.168.193.0/24     0.0.0.0/0
    0     0 LOGDROP    all  --  *      *       1.0.0.0/8            0.0.0.0/0
    0     0 LOGDROP    all  --  *      *       2.0.0.0/8            0.0.0.0/0
    0     0 LOGDROP    all  --  *      *       5.0.0.0/8            0.0.0.0/0
    0     0 LOGDROP    all  --  *      *       7.0.0.0/8            0.0.0.0/0
    0     0 LOGDROP    all  --  *      *       10.0.0.0/8           0.0.0.0/0
    0     0 LOGDROP    all  --  *      *       23.0.0.0/8           0.0.0.0/0
    0     0 LOGDROP    all  --  *      *       27.0.0.0/8           0.0.0.0/0
```

```
    0     0 LOGDROP    all  --  *      *       31.0.0.0/8          0.0.0.0/0
    0     0 LOGDROP    all  --  *      *       36.0.0.0/7          0.0.0.0/0
    0     0 LOGDROP    all  --  *      *       39.0.0.0/8          0.0.0.0/0
    0     0 LOGDROP    all  --  *      *       41.0.0.0/8          0.0.0.0/0
    0     0 LOGDROP    all  --  *      *       42.0.0.0/8          0.0.0.0/0
    0     0 LOGDROP    all  --  *      *       49.0.0.0/8          0.0.0.0/0
    0     0 LOGDROP    all  --  *      *       50.0.0.0/8          0.0.0.0/0
    0     0 LOGDROP    all  --  *      *       58.0.0.0/7          0.0.0.0/0
    0     0 LOGDROP    all  --  *      *       70.0.0.0/7          0.0.0.0/0
    0     0 LOGDROP    all  --  *      *       72.0.0.0/5          0.0.0.0/0
    0     0 LOGDROP    all  --  *      *       83.0.0.0/8          0.0.0.0/0
    0     0 LOGDROP    all  --  *      *       84.0.0.0/6          0.0.0.0/0
    0     0 LOGDROP    all  --  *      *       88.0.0.0/5          0.0.0.0/0
    0     0 LOGDROP    all  --  *      *       96.0.0.0/7          0.0.0.0/0
    0     0 LOGDROP    all  --  *      *       169.254.0.0/16      0.0.0.0/0
    0     0 LOGDROP    all  --  *      *       172.16.0.0/12       0.0.0.0/0
    0     0 LOGDROP    all  --  *      *       173.0.0.0/8         0.0.0.0/0
    0     0 LOGDROP    all  --  *      *       174.0.0.0/7         0.0.0.0/0
    0     0 LOGDROP    all  --  *      *       176.0.0.0/5         0.0.0.0/0
    0     0 LOGDROP    all  --  *      *       184.0.0.0/6         0.0.0.0/0
    0     0 LOGDROP    all  --  *      *       189.0.0.0/8         0.0.0.0/0
    0     0 LOGDROP    all  --  *      *       190.0.0.0/8         0.0.0.0/0
    0     0 LOGDROP    all  --  *      *       192.0.2.0/24        0.0.0.0/0
    0     0 LOGDROP    all  --  *      *       192.168.0.0/16      0.0.0.0/0
    0     0 LOGDROP    all  --  *      *       197.0.0.0/8         0.0.0.0/0
    0     0 LOGDROP    all  --  *      *       223.0.0.0/8         0.0.0.0/0

Chain LOGDROP (38 references)
 pkts bytes target     prot opt in     out     source              destination
    0     0 LOG        all  --  *      *       0.0.0.0/0           0.0.0.0/0           LOG flags 6 level 4
      prefix `www-a'
    0     0 DROP       all  --  *      *       0.0.0.0/0           0.0.0.0/0

Chain VALID (1 references)
 pkts bytes target     prot opt in     out     source              destination
    0     0 ACCEPT     udp  --  *      *       0.0.0.0/0           192.168.65.31       udp dpt:694
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0           192.168.65.31       tcp dpt:80
    0     0 ACCEPT     udp  --  *      *       192.168.65.21       192.168.65.31       udp dpt:123
    0     0 ACCEPT     udp  --  *      *       192.168.65.22       192.168.65.31       udp dpt:123
    0     0 ACCEPT     udp  --  *      *       192.168.65.21       192.168.65.31       udp dpt:53
    0     0 ACCEPT     udp  --  *      *       192.168.65.22       192.168.65.31       udp dpt:53
```

```
0     0 ACCEPT     esp  --  *      *       192.168.193.0/24     192.168.65.31
0     0 ACCEPT     ah   --  *      *       192.168.193.0/24     192.168.65.31
0     0 ACCEPT     udp  --  *      *       192.168.193.0/24     192.168.65.31      udp spt:500 dpt:500
0     0 ACCEPT     tcp  --  *      *       192.168.193.11       192.168.65.31      tcp dpt:22
```

Appendix F: DMZ Switch Configuration

The following is a completed configuration file for one of the Cisco Catalyst 3550 DMZ switches. This file is meant to be transferred to the router by TFTP on initial configuration, and therefore has additional commands and comments that will not necessarily be present in the "show running-config" output after the router is configured. Cisco does not store comments in router online configurations, and some of the commands are reinforcing defaults in case it becomes necessary to load the configuration on a version of IOS other than 12.3.

```
! Verbose logging timestamps
service timestamps debug datetime msec localtime show-timezone
service timestamps log datetime msec localtime show-timezone
! Encrypt all passwords
service password-encryption
no service dhcp
no service udp-small-servers
no service tcp-small-servers
!
hostname GIACE-DMZ-1
!
logging buffered 16384 notifications
! Crackers... get a life!  Stronger passwords will be used in the final
      config.
enable secret 5 $1$uL49$uHdSDpaPfG7dAHdw/UsJA.
!
username GIACEuser password 7 1515040D132B3265
!
clock timezone EST -5
no ip source-route
no ip finger
no ip domain-lookup
!
no ip bootp server
!
interface VLAN1
 description Internal Interface
 ip address 192.168.65.250
 no ip route-cache
 no ip unreachables
 arp timeout 0
!
interface FastEthernet0/1
 description Firewall_Default Gateway
 port security max-mac-count 1
 port security action trap
 spanning-tree portfast
 no cdp enable
!
interface FastEthernet0/2
 description IDS
 port security max-mac-count 1
```

```
 port security action trap
 spanning-tree portfast
 no cdp enable
! The following will copy all frames from Fast0/1 to this port
! This will allow the IDS to see all inbound attacks, but will
! not see inter-host attacks on the DMZ
 port monitor FastEthernet 0/1
!
interface FastEthernet0/3
 description Informational Web Server 1
 port security max-mac-count 1
 port security action trap
 spanning-tree portfast
 no cdp enable
!
interface FastEthernet0/4
 description Informational Web Server 2
 port security max-mac-count 1
 port security action trap
 spanning-tree portfast
 no cdp enable
!
interface FastEthernet0/5
 description Ecommerce Web Server 1
 port security max-mac-count 1
 port security action trap
 spanning-tree portfast
 no cdp enable
!
interface FastEthernet0/6
 description Ecommerce Web Server 2
 port security max-mac-count 1
 port security action trap
 spanning-tree portfast
 no cdp enable
!
interface FastEthernet0/7
 description Proxy 1
 port security max-mac-count 1
 port security action trap
 spanning-tree portfast
 no cdp enable
!
interface FastEthernet0/8
 description Proxy 2
 port security max-mac-count 1
 port security action trap
 spanning-tree portfast
 no cdp enable
!
interface FastEthernet0/9
 description Certficate Server
 port security max-mac-count 1
 port security action trap
 spanning-tree portfast
 no cdp enable
!
```

```
interface FastEthernet0/10
 shutdown
 port security max-mac-count 1
 port security action trap
 spanning-tree portfast
 no cdp enable
!
interface FastEthernet0/11
 shutdown
 port security max-mac-count 1
 port security action trap
 spanning-tree portfast
 no cdp enable
!
interface FastEthernet0/12
 shutdown
 port security max-mac-count 1
 port security action trap
 spanning-tree portfast
 no cdp enable
!
interface FastEthernet0/13
 shutdown
 port security max-mac-count 1
 port security action trap
 spanning-tree portfast
 no cdp enable
!
interface FastEthernet0/14
 shutdown
 port security max-mac-count 1
 port security action trap
 spanning-tree portfast
 no cdp enable
!
interface FastEthernet0/15
 shutdown
 port security max-mac-count 1
 port security action trap
 spanning-tree portfast
 no cdp enable
!
interface FastEthernet0/16
 shutdown
 port security max-mac-count 1
 port security action trap
 spanning-tree portfast
 no cdp enable
!
interface FastEthernet0/17
 shutdown
 port security max-mac-count 1
 port security action trap
 spanning-tree portfast
 no cdp enable
!
interface FastEthernet0/18
```

```
 shutdown
 port security max-mac-count 1
 port security action trap
 spanning-tree portfast
 no cdp enable
!
interface FastEthernet0/19
 shutdown
 port security max-mac-count 1
 port security action trap
 spanning-tree portfast
 no cdp enable
!
interface FastEthernet0/20
 shutdown
 port security max-mac-count 1
 port security action trap
 spanning-tree portfast
 no cdp enable
!
interface FastEthernet0/21
 shutdown
 port security max-mac-count 1
 port security action trap
 spanning-tree portfast
 no cdp enable
!
interface FastEthernet0/22
 shutdown
 port security max-mac-count 1
 port security action trap
 spanning-tree portfast
 no cdp enable
!
interface FastEthernet0/23
 shutdown
 port security max-mac-count 1
 port security action trap
 spanning-tree portfast
 no cdp enable
!
interface FastEthernet0/24
 shutdown
 port security max-mac-count 1
 port security action trap
 spanning-tree portfast
 no cdp enable
!
no ip http server
no ip http server-secure
!
! send copious logs to syslog
logging trap debugging
logging facility local5
! firewall will nat logs to the logging server
logging host 1.2.3.2
!
```

```
no cdp run
snmp-server engineID local 0000000902000002166 47FB5
snmp-server community GIACE-strong-snmp RO 20
snmp-server enable traps snmp
snmp-server enable traps hsrp
snmp-server enable traps config
snmp-server enable traps syslog
! server to send SNMP traps to - again, firewall will NAT in...
snmp-server host 1.2.3.2 dmz-switches
!
banner motd #

Switch GIACE-DMZ-1.  Access to this device or attached networks by
unauthorized users is prohibited.  No user of this device or connected
networks has any expectation of privacy.
Violators will be prosecuted.

#
!
line con 0
 transport input none
line aux 0
 transport input none
 transport output none
line vty 0 4
 no login
 transport input none
 transport output none
!
! setup ntp in authentication mode - go in to DMZ servers
ntp authentication-key 1 md5 09107D2C3A3732263427211375 7
ntp authenticate
ntp server 1.2.3.22
ntp server 1.2.3.21 prefer
end
```

Bibliography

Bastien, Greg & Degu, Christian Abera. CCSP Cisco Secure PIX Firewall Advanced Exam Certification Guide. 2003. Cisco Press: Indianapolis, IN.

Brenton, Chris; et al. Track 2 – Firewalls, Perimeter Protection and VPNs: 2.4 Defense In-Depth. 2003.

Brenton, Chris; et al. Track 2 – Firewalls, Perimeter Protection and VPNs: 2.5 VPNs. 2003.

Center for Internet Security, The. "Linux Benchmark v1.1.0." http://www.cisecurity.org/tools2/linux/LinuxBenchmark.pdf. (December 29, 2003).

Cisco Systems, Inc. "Cisco IOS Software Releases 12.3 Mainline Command References." http://www.cisco.com/en/US/products/sw/iosswrel/ps5187/prod_command_reference_list.html. (December 29, 2003)

Cisco Systems, Inc. "Cisco PIX Firewall Command Reference, Version 6.3." http://www.cisco.com/en/US/products/sw/secursw/ps2120/products_command_reference_book09186a008017284e.html. (December 29, 2003)

Cisco Systems, Inc. "Cisco PIX Firewall Software Managing VPN Remote Access." http://www.cisco.com/en/US/products/sw/secursw/ps2120/products_configuration_guide_chapter09186a0080172787.html. (December 29, 2003)

Cisco Systems, Inc. "Configuration Examples Related to VLAN Features." Sep. 29, 2003. http://www.cisco.com/univercd/cc/td/doc/product/lan/28201900/1928v8x/eescg8x/aleakyv.htm. (December 29, 2003)

Cisco Systems, Inc. "Using Cisco PIX Firewall Failover." http://www.cisco.com/en/US/products/sw/secursw/ps2120/products_configuration_guide_chapter09186a008017278a.html. (December 29, 2003)

Dittrich, David. "The DoS Project's 'trinoo' distributed denial of service attack tool." Oct. 21, 1999. http://staff.washington.edu/dittrich/misc/trinoo.analysis.txt. (December 29, 2003)

Fyodor. "Nmap: The Art of Port Scanning." Sep. 6, 1997. http://www.insecure.org/nmap/nmap_doc.html. (December 29, 2003)

Hunt, Craig. TCP/IP Network Administration. 1992. O'Reilly & Associates, Inc.: Sebastopol, CA.

ISS. "OpenSSH Could Allow an Attacker to Obtain Valid Administrative Account." May 1, 2003. http://xforce.iss.net/xforce/xfdb/11970. (December 29, 2003)

ISS. "OpenSSH Large Packet Buffer Overflow." Sep. 16, 2003. http://xforce.iss.net/xforce/xfdb/13191. (December 29, 2003)

Lammle, Todd & Hales, Kevin. CCNP Switching Study Guide. 2001. Sybex: San Francisco.

Microsoft. "Microsoft Security Bulletin MS03-033: Unchecked Buffer in MDAC Function Could Enable System Compromise (823718)." Aug. 20, 2003. http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS03-033.asp. (December 29, 2003)

Miller, Timothy. "SANS GIAC Firewall Analyst Practical Assignment (GCFW)." July 11, 2003. http://www.giac.org/practical/GCFW/Timothy_Miller_GCFW.pdf. (December 16, 2003.)

Novak, Judy; et al. Track 2 – Firewalls, Perimeter Protection and VPNs: 2.1 TCP/IP for Firewalls. 2003.

Tcpdump.org. "TCPDUMP (8) man page." Jan. 3, 2001. "man tcpdump" (December 16, 2003.)

Thomas, Rob. "Secure IOS Template v3.1." Nov. 17, 2003. http://www.cymru.com/Documents/secure-ios-template.html. (December 29, 2003)