Global Information Assurance Certification Paper

SANS GIAC Training & Certification

Level Two

Firewalls, Perimeter Protection, and VPNs

GCFW Practical Assignment

Chris Olson

March 30, 2001

Assignment 1 – Security Architecture

Assignment Text:

Define a security architecture for GIAC Enterprises, a growing Internet startup that expects to earn $200 million per year in online sales of fortune cookie sayings, and which has just completed a merger/acquisition. Your architecture must specify filtering routers, firewalls, VPNs to partners, secure remote access, and internal firewall. Be explicit about the brand and version of each perimeter defense component. Produce a diagram or set of diagrams with explanatory text that define how to use perimeter technologies to implement your security architecture.

You must consider and define access for:

- Customers (the companies that purchase bulk online fortunes);
- Suppliers (the authors of fortune cookie sayings that connect to supply fortunes);
- Partners (the international partners that translate and resell fortunes).

Solution:

We'll start with the assumption that the default security policy is to "deny all" unless there is a documented business need for access to be granted. This leads us to the initial tasks of determining what business needs exist. When service and access needs have been established, we can develop our security architecture around those needs.

Services required

- Internet
  Any business connected to the internet these days must have access to some basic services such as DNS, email, and web browsing. Many companies, in addition to being consumers of these services, also own their own domain name, so put themselves in the business of providing services as well. It was decided that that DNS, web, and email administration will be provided in-house instead of outsourcing. This will allow for a greater level of control and auditing than if they were outsourced.
  - **DNS** – We need to provide primary SOA service for the addresses in the giacenterprises.com domain which we want to be externally visible.
  - **Web** – We need to provide a server for www.giacenterprises.com so people can learn about the company, and potential new customers can get information regarding the products and services that are offered.
  - **Email** – There needs to be server which email for the giacenterprises.com domain will be sent. This will also be used as an email storage spot until employees access it with their mail clients.
- Customers

- **Customer data** – Customers need the ability to make secure transactions in order to obtain the product. For GIAC Enterprises, the product is text information, so it is ideally provided via the internet. Because the sole product is information which is transferred over the internet, it is critical that this be kept secure.

- Suppliers
  - **Supplier data** – Suppliers also need the ability to make secure transactions in order to upload their product. The supplier data needs to be equally secure as the customer data.
- Partners

  Because of the higher level of access required by partners, particular attention will be paid to the security and integrity of the access given to them.
  - **Full database access** – In addition to being able to access the data provided to customers and suppliers, business partners need an additional level of access. They need full backend database access, which they can generate their own reports and queries.
  - **Fileserver access** – Partners also need access to the fileserver for bookkeeping, accounting, documentation, etc.
- Security Services
  - **Intrusion detection** – A significant number of services are being provided over the public internet. This requires that security policies be open enough to allow these services to function properly. Intrusion Detection Systems (IDS) will be implemented on the subnets which are the most vulnerable.
  - **Syslog collection** – A centralized point for logging and exception monitoring. Syslog collection also helps prevent intruders from covering their tracks in the event of a breakin. If the logs are being constantly sent to a remote server, there is a much better chance of detecting a cracker who scrubs the logs on the local machine in an attempt to hide their tracks.
  - **Virtual Private Network access** – A VPN is not really a service itself, but rather a solution used to provide other services. I mention it here because VPNs are becoming so common that many people do consider it a standard service. In a nutshell, a VPN may be used by partners, sales staff, and telecommuters to gain access to services that are normally blocked by the firewall(s).
- Internal Services

  These are services which are required by the GIAC Enterprises internal staff. They include things that are needed for running the business, in addition to IT infrastructure services.
  - **Database** – This is the master database that holds all of the fortune cookie data used by GIAC, it's customers, suppliers, and partners. It should be designed for security and high availability.
  - **Fileserver** - Fileserver access should be granted on a departmental and need basis.

- **Backups** – All servers will need to have their data backed up on a regular basis. Backups will be done over the network to dedicated backup servers with attached tape libraries.
- **DNS** – Internal DNS is needed to provide full name service for the domain. Unlike the external DNS server mentioned above, the internal server will contain name records for all nodes, both internal and external.

Access required

Internet – The internet services listed above (DNS, web, and email) need to be available from anywhere on the internet.

- DNS – port 53 UDP for DNS queries from anywhere. If there are secondary DNS servers which are hosted elsewhere, then port 53 TCP will also need to be permitted from those secondaries.
- Web – TCP port 80 for regular HTTP queries, and TCP port 443 for HTTP over SSL will need to be allowed to the web server.
- Email – TCP port 25 connections will need to be allowed for SMTP transfers to the mail server

Customers & Suppliers – To allow customers and suppliers to securely transfer fortune data from and to GIAC Enterprises servers, TCP port 443 (HTTP over SSL) and TCP port 22 (ssh and scp) will be permitted to the appropriate servers.

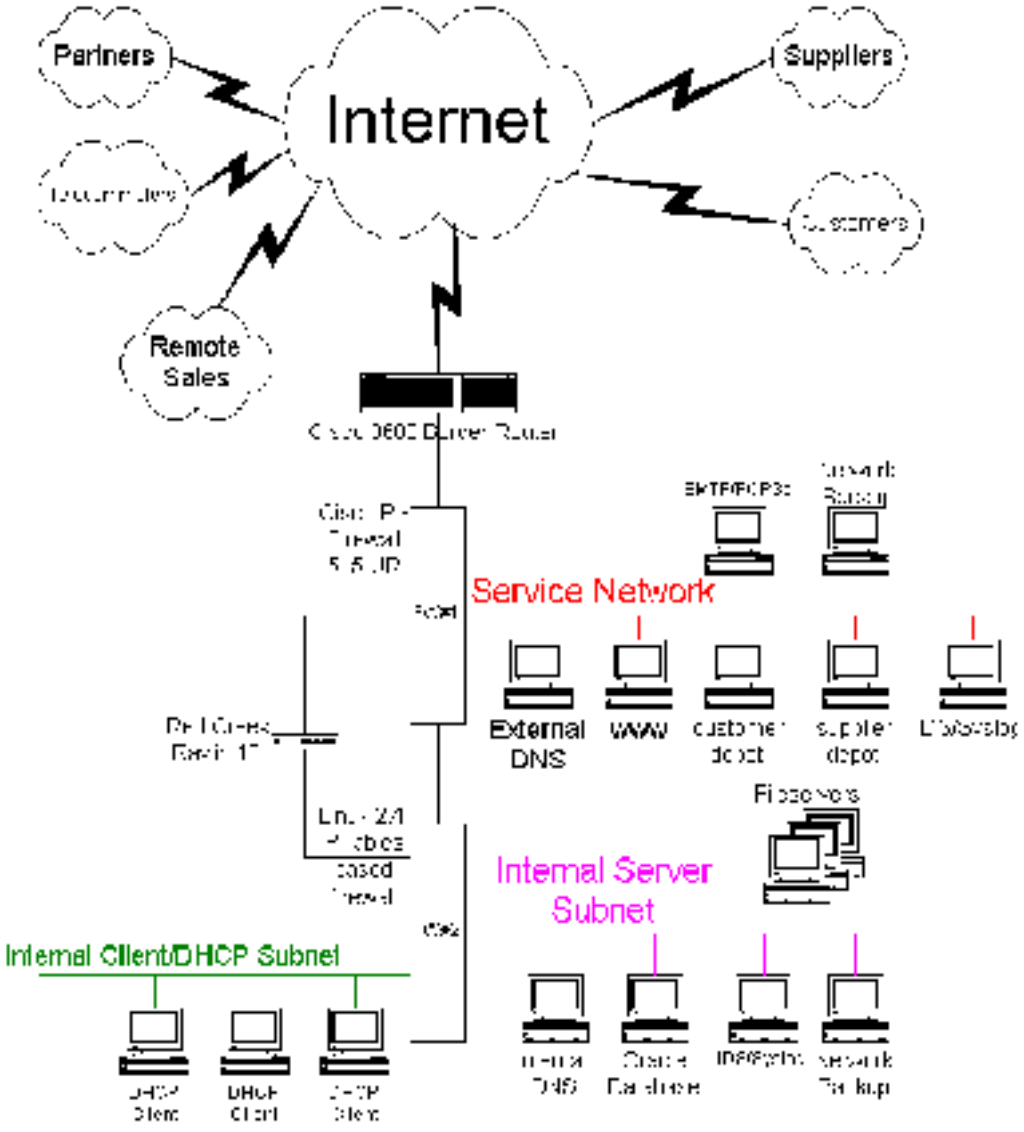
Partners – because partners will need access to a significant number of backend services, their access will best be provided with a VPN solution instead of and ACL type solution. Partners need full access to the backend Oracle database and also to some of the departmental fileserver data.

Security Services – Security services such as IDS and syslogging can and should be done in a localized way such that the data collected does not need to traverse security barriers such as firewalls and routers. This will mean that multiple servers need to be constructed.

Internal Services

- Internal services like Oracle database access and departmental fileservers will need to be access by many or all of the internal company staff. This access will primarily come from DHCP desktop clients, but there is also be a need to allow telecommuters and mobile sales staff access to this data via the VPN.
- Email – Employees email will be fetched from the internet SMTP server mentioned above. Only POP3 over SSL (TCP port 995) and IMAP over SSL (TCP port 993) will be allowed as ways to fetch mail from this server. This will allow secure mail fetching from both internal and external (sales staff) locations.

Security Design



Design Overview

The security design above was done with several things in mind: separation of services, practical implementation, and defense in depth.

The primary design consideration for this architecture was "who needs access to what." Based on the service definitions and access requirements listed above, the computers at GIAC Enterprises will split into three groups.

1. **External services**. This group of machines either provides services to the entire internet (www, DNS, etc.) or to a large number of customers and suppliers. The customers will have a dedicated depot machine setup in the service network. This machine will not hold the production database, but will instead be fed by the

backend Oracle database as needed. Customers will be allowed to access their depot server securely via HTTP over SSL or ssh/scp as needed. Suppliers will be provided an analogous depot machine for their needs. Servers are limited to a single function whenever practical. This limits the amount of data that a cracker would have access to if they were able to compromise a single server in the service network. There will also be two administrative type machines on the service network: a network backup server, and an IDS/syslog server. These machines are necessary for the data integrity and security of the machines on this subnet, but should not be accessed from the internet. There is some amount of risk involved with putting the customer and supplier depots on the same subnet as the general purpose internet servers, but this risk was deemed acceptable. Access control on the firewall, combined with placing an intrusion detection system on the service network, should provide adequate security for this situation. Because this subnet presents the greatest risk, these servers should be watched closely. The OS of each machine should be hardened as much as possible, and security audits should be performed frequently.

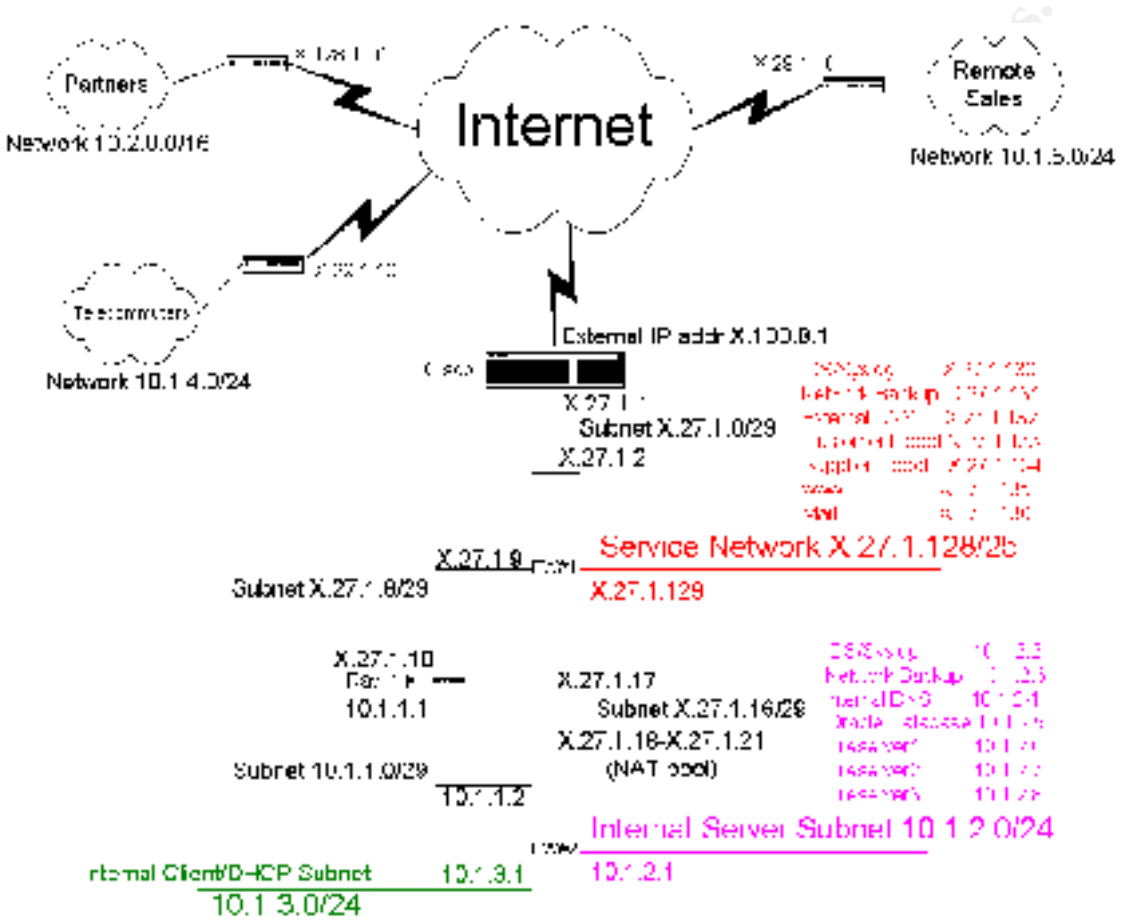
2. **Internal services**. This is where the main servers at GIAC Enterprises will go. These machines will include the master Oracle fortune database, internal DNS, and departmental fileservers. Machines on this subnet may need to be accessed by on-site staff and administrators. Some of these servers will also need to be accessed by partners, sales staff, and telecommuters via the VPN. The IP addresses on this subnet will be private (RFC1918). For external access these addresses will be NATed to a small pool of static addresses at firewall #2. The communication between the internal services subnet, and the DHCP subnet will not be NATed.
3. **Corporate desktops**. These are primarily desktop and laptop workstations used by on-site staff. They will be assigned addresses via DHCP. These machines are strictly clients, so they should not be accessed from anywhere. The IP addresses on this subnet will be private (RFC1918). For external access these addresses will be NATed to a small pool of static addresses at firewall #2.

The VPN configuration was probably the most difficult design decision to make. There is a strong argument to be made that the IPSec VPN capabilities of the Pix firewall should be used instead of buying a separate Red Creek VPN box. But because there are still significant interoperability issues between different vendors implementation of IPSec, whatever VPN solution was decided upon, our partners, remote sales staff and telecommuters would all need to use a single vendor solution. Partner's connectivity poses the largest issue because they already have their own infrastructure established. If a Cisco Pix VPN was used, that would require the partners to also have the same firewall on their side, which is not a good assumption to make. For this reason, the Red Creek Ravlin 10 was chosen. It should be far easier to insert another Red Creek Ravlin box into the partner's network than it would be to potentially switch their firewall vendor. Remote sales staff can use the RavlinSoft software for access, and telecommuters can either use RavlinSoft or a Personal Ravlin depending upon their operating systems and uplink speed. Another benefit of using a separate box for VPN services is that it reduces the load that would otherwise be placed on the firewall. The Ravlin 10 model is fast

enough to encrypt/decrypt at 10BaseT "wire speed". This means that it should be plenty fast enough to handle as many security associations from partners, sales staff, and telecommuters as required and not create a bandwidth bottleneck. The bottleneck should be and will be the speed of the uplink. The placement of the Ravlin was another consideration. The box was intentionally placed between the two firewalls for greater traffic control. By feeding the VPN from firewall #1, we are able to control the source and protocol of connections to the Ravlin. By placing firewall #2 downstream from the Ravlin, we are able to limit which backend servers the other end of the VPN has access to. For example, access via the VPN tunnels, should not be allowed to connect to the DHCP clients or to the security and backup servers on the server subnet.

Defense in depth is enhanced by using three different devices: a Cisco border router IOS 12.1 which will connect us to the outside world and do basic packet filtering, a Cisco Pix firewall version 5.3 to which we'll connect our Service Network, and a Debian GNU/Linux 2.4.3 IPTables firewall in back to connect our back end servers and workstations. The two backend subnets will be assigned private IP address space, and the linux box will use NAT when communicating to firewall #1. Separate firewalls from two different vendors were intentionally chosen to reduce the risk from vendor specific exploits. The downside of this product diversity is training. The security engineers will need to be familiar with all of the products involved. There is also some risk associated with using the new linux 2.4 IPTables solution instead of the older 2.2 series kernel with IPchains. Because 2.4 is so new, it has not been as thoroughly tested as 2.2 has and could present us with unknown stability or security challenges in the future. These risks were deemed acceptable because the new features of 2.4. Namely, 2.4 has the ability to do statefull firewalling, as opposed to stateless packet filtering that 2.2 did. 2.4 also has the ability to use multiple addresses for it's NAT pool. 2.2 could only do single address masquerading. Further, because this Linux firewall is the secondary, backend firewall, it will be far less exposed to potential exploits than the Cisco Pix will be. Anyhow, it would be wise for the security team to pay particular attention to the developments of Linux 2.4 and IPTables.

IP Addressing



Assignment 2 – Security Policy

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

- Border Router
- Primary Firewall
- VPN

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

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

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

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

1. The service or protocol addressed by the ACL or rule, and the reason these services might be considered a vulnerability.
2. Any relevant information about the behavior of the service or protocol on the network.
3. The syntax of the ACL, filter, rule, etc.
4. A description of each of the parts of the filter.
5. An explanation of how to apply the filter.
6. If the filter is order-dependent, list any rules that should precede and/or follow this filter, and why this order is important. (Note: instead of explaining order dependencies for each individual rule, you may wish to create a separate section of your practical that describes the order in which ALL of the rules should be applied, and why.)
7. Explain how to test the ACL/filter/rule.

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

Solution:

- **Border Router**
  The purpose of the border router is to filter out "the easy stuff". These are the basic ACLs that should be on any border router regardless of services behind it. This will reduce the load on the primary firewall, and add to our goal of defense in depth. Because any traffic passing through our router will encounter the Checkpoint firewall immediately, the philosophy of "deny all except what is explicitly needed" will not be followed for the border router security policy. Instead, the restrictive rules allowing only specific traffic will be implemented on the firewall. Duplicating those rules on the border router, would not make the environment more secure. It would only add complexity, and extra work when making a change. So the border router will be used to add to the functionality of the firewall, not duplicate it.

  A note about testing rules: The following is a specific listing of rules, ACLs, and configuration information for our environment. There are three primary tests which apply to all rules
  1. Does the service involved function as expected.
  2. An audit will be performed immediately following this design. The audit may show unexpected vulnerabilities.
  3. If anything is discovered in #1 or #2, diagnostic tools like tcpdump, ethereal, nmap, and hping2, along with router and firewall log analysis can be used to troubleshoot any problems that arise.

  Border Router (Cisco 3600 IOS 12.1)

```
# don't store router passwords in cleartext
service password-encryption
# prevent smurf attacks
no ip directed-broadcast
# don't give out MAC addresses
no ip proxy-arp
# don't allow reverse mapping with ICMP info
no ip unreachables
# don't allow router to provide services except routing
no service finger
no service udp-small-servers
no service tcp-small-servers
no ip bootp server
no ip http server
no ntp master
# don't allow fancy routing
no ip source-route
# log everything to the syslog server in the service network
logging X.27.1.130
# go away banner
banner / WARNING: Authorized Access Only /
```

Ingress ACL

```
# don't allow inbound packets with illegal source addresses
access-list 100 deny host 0.0.0.0 any log
access-list 100 deny 127.0.0.0 0.255.255.255 any log
access-list 100 deny 10.0.0.0 0.255.255.255 any log
access-list 100 deny 172.16.0.0 0.31.255.255 any log
access-list 100 deny 192.168.0.0 31.255.255.255 any log
access-list 100 deny 255.0.0.0 0.255.255.255 any log
access-list 100 deny 224.0.0.0 31.255.255.255 any log

# only allow certain inbound ICMP
access-list 100 permit icmp any X.27.1.0 0.0.0.255 0 # echo reply
access-list 100 permit icmp any X.27.1.0 0.0.0.255 3 # unreachable
access-list 100 permit icmp any X.27.1.0 0.0.0.255 4 # source quench
access-list 100 permit icmp any X.27.1.0 0.0.0.255 11 # time exceeded
access-list 100 deny icmp any any log

# allow everything else, with the idea that most of the heavy lifting
# will be done on the firewalls
access-list 100 permit any any
```

Egress ACL

```
# don't allow any packets to leave our network with improper source
addresses
access-list 101 deny host 0.0.0.0 any log
access-list 101 deny 127.0.0.0 0.255.255.255 any log
access-list 101 deny 10.0.0.0 0.255.255.255 any log
access-list 101 deny 172.16.0.0 0.31.255.255 any log
access-list 101 deny 192.168.0.0 31.255.255.255 any log
access-list 101 deny 255.0.0.0 0.255.255.255 any log
access-list 101 deny 224.0.0.0 31.255.255.255 any log
# only permit ICMP echo request packets for diagnostics
access-list 101 permit icmp X.27.1.0 0.0.0.255 any 8
access-list 101 deny icmp any any log
# allow everything else, with the idea that most of the heavy lifting
# will be done on the firewalls
access-list 101 permit any any
```

• Firewall #1 (Cisco Pix 5.3)

The cisco pix firewall is a statefull firewall. It works by assigning security levels (a number between 0 and 100) to each one. 0 is the lowest security level and is assigned to the outside interface. 100 is the highest and is assigned to the inside interface. For routing from a lower security level interface to a higher security level interface, all traffic is denied by default, and must be explicitly allowed by ACLsif needed. For routing from a higher security level interface to a lower security level interface, all traffic is allowed by default, and must be explicitly denied with ACLs if needed.

```
# name the interfaces and assign security levels to them
# outside goes to the border router
# service is connected to the service network
# vpn goes to the Ravlin10
# inside goes to firewall #2
nameif ethernet0 outside security0
nameif ethernet1 service security25
```

```
nameif ethernet2 vpn security50
nameif ethernet3 inside security100

# Assign IP addresses and subnet masks
ip address outside X.27.1.2 255.255.255.248
ip address service X.27.1.129 255.255.255.128
ip address vpn X.27.1.9 255.255.255.248
ip address inside X.27.1.17 255.255.255.248

# set hostname and misc settings
hostname firewall1
arp timeout 14400
no failover
names
pager lines 24
logging buffered debugging
mtu outside 1500
mtu service 1500
mtu vpn 1500
mtu inside 1500

# Turn off RIP on all interfaces except the inside
rip inside passive
no rip outside passive
no rip dmz passive
no rip vpn passive
no rip inside default
no rip outside default
no rip dmz default
no rip vpn default

# Set default route
route outside 0.0.0.0 0.0.0.0 X.27.1.1 1

# setup ACL names for inbound traffic
access-group acl_outside in interface outside
access-group acl_service in interface service
access-group acl_vpn in interface vpn
access-group acl_inside in interface inside

# setup name shortcuts, so we don't have to remember the IP
addresses
name X.27.1.1 border_router
name X.27.1.18 firewall2
name X.27.1.10 ravlin
name X.27.1.130 syslog_server
name X.27.1.131 backup_server
name X.27.1.132 dns_server
name X.27.1.133 cust_depot
name X.27.1.134 supp_depot
name X.27.1.135 web_server
name X.27.1.136 mail_server
```

```
name X.128.1.10 partner_ravlin
name X.29.1.10 soho_ravlin
name X.30.1.10 sales_ravlin
```

inside

The only rules we setup on the inside interface is the basic egress ACL. Only traffic that is from the small subnet between the two firewalls is permitted. Anything else is assumed to be spoofed and is blocked. Most of the internal rules are taken care of by firewall #2.

```
access-list acl_inside permit ip X.27.1.16 255.255.255.248
access-list acl_inside deny ip any any
```

VPN

Our VPN solution uses ISAKMP for authentication and ESP for encapsulation. ISAKMP uses UDP port 500 at each end of the connection and ESP uses IP protocol 50. The following rules restrict the traffic to only those two protocols between the Ravlin 10 and the known addresses of the remote Ravlin boxes belonging to the parter, SOHO (telecommuter), and remote sales. Because the VPN box provides a path behind both firewalls, everything else is blocked except what is absolutely required.

```
access-list acl_outside permit udp host partner_ravlin eq 500 host ravlin eq 500
access-list acl_outside permit udp host soho_ravlin eq 500 host ravlin eq 500
access-list acl_outside permit udp host sales_ravlin eq 500 host ravlin eq 500
access-list acl_outside permit 50 host partner_ravlin host ravlin
access-list acl_outside permit 50 host soho_ravlin host ravlin
access-list acl_outside permit 50 host sales_ravlin host ravlin

access-list acl_vpn permit udp host ravlin eq 500 partner_ravlin eq 500
access-list acl_vpn permit udp host ravlin eq 500 soho_ravlin eq 500
access-list acl_vpn permit udp host ravlin eq 500 sales_ravlin eq 500
access-list acl_vpn permit 50 host ravlin partner_ravlin
access-list acl_vpn permit 50 host ravlin soho_ravlin
access-list acl_vpn permit 50 host ravlin sales_ravlin
access-list acl_vpn deny ip any any
```

service

Traffic coming from the outside to the service network is limited to only the ports needed for the services provided.

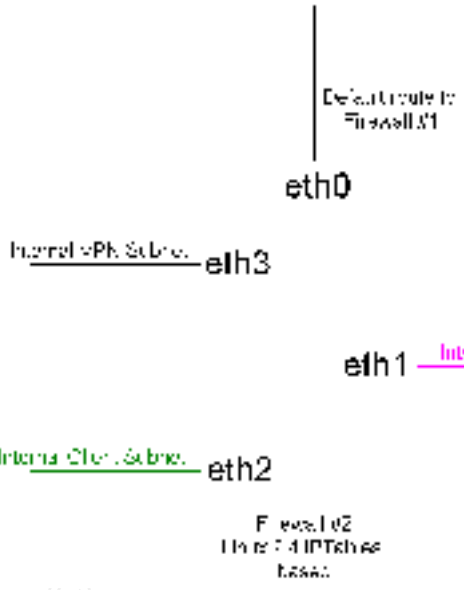
```
# allow the border router to use the syslog
# server in the service network
access-list acl_outside permit udp border_router syslog eq 514

# allow DNS queries (udp 53 ) to the DNS server, but
# not zone transfers (TCP 53)
access-list acl_outside permit udp any dns_server eq 53

# allow ssh and HTTPs access to the customer and supplier depots.
# For added security, this access could be restricted to originate
# only from predefined IP addresses, but at this time that level
```

```
# did not seem necessary.
access-list acl_outside permit tcp any cust_depot eq 22
access-list acl_outside permit tcp any cust_depot eq 443
access-list acl_outside permit tcp any supp_depot eq 22
access-list acl_outside permit tcp any supp_depot eq 443

# allow HTTP and HTTPS traffic to the public web server
access-list acl_outside permit tcp any web_server eq 80
access-list acl_outside permit tcp any web_server eq 443

# allow SMTP, SMTP over SSL and POP3 over SSL into the mail server
access-list acl_outside permit tcp any mail_server eq 25
access-list acl_outside permit tcp any mail_server eq 465
access-list acl_outside permit tcp any mail_server eq 995
```

With the exception of the mail server relaying SMTP and SMTP over SSL, machines on the service network should not initiate connections to anywhere. A deny all entry is added to the end to override the default allow to lower security levels.

```
access-list acl_service permit tcp mail_server eq 25 any
access-list acl_service permit tcp mail_server eq 465 any
access-list acl_service deny ip any any
```

- **Firewall #2 (Intel Linux 2.4.3 using IPTables)**




```
# Turn on IP forwarding
echo 1 > /proc/sys/net/ipv4/ip_forward

# the interfaces have been assigned the following IP addresses at bootup
# outside eth0 X.27.1.18 (X.27.1.19 - X.27.1.21 will be added below)
# server eth1 10.1.2.1
# client eth2 10.1.3.1
# vpn eth3 10.1.1.2

# add 3 additional IP addresses to the outside interface to be used as a
NAT pool
```

```
ip address add X.27.1.19 dev eth0
ip address add X.27.1.20 dev eth0
ip address add X.27.1.21 dev eth0

# configure the outside interface to NAT all traffic going out to the
pool of four addresses
iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to X.27.1.18-X.27.1.21

# setup four new chains to be used for traffic incoming to that interface
iptables -N in-eth0
iptables -N in-eth1
iptables -N in-eth2
iptables -N in-eth3
iptables -A INPUT -i eth0 -j in-eth0
iptables -A INPUT -i eth1 -j in-eth1
iptables -A INPUT -i eth2 -j in-eth2
iptables -A INPUT -i eth3 -j in-eth3
iptables -A FORWARD -i eth0 -j in-eth0
iptables -A FORWARD -i eth1 -j in-eth1
iptables -A FORWARD -i eth2 -j in-eth2
iptables -A FORWARD -i eth3 -j in-eth3
```

outside

```
# allow established connections.  This line makes our firewall
# statefull.  Deny everything else.  No incoming new connections
# should be allowed at all.
iptables -A in-eth0 -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A in-eth0 -j DROP
```

server

```
# allow established connections in a statefull way
iptables -A in-eth1 -m state --state ESTABLISHED,RELATED -j ACCEPT

# allow ssh connections from the Oracle database to the
# customer and supplier depot servers in the service network.
# The Oracle database server will use scp to automatically sync
# up with the depot servers, keeping the data stored on those
# depots to a minimum.
iptables -A in-eth1 -p TCP -s 10.1.2.5 -d X.27.1.133 --dport 22 -j ACCEPT
iptables -A in-eth1 -p TCP -s 10.1.2.5 -d X.27.1.134 --dport 22 -j ACCEPT

# allow any server in this subnet to relay SMTP email through
# the mail server in the service network.
iptables -A in-eth1 -p TCP -s 10.1.2.0/24 -d X.27.1.136 --dport 25 -j
ACCEPT
iptables -A in-eth1 -p TCP -s 10.1.2.0/24 -d X.27.1.136 --dport 465 -j
ACCEPT
# drop everything else.  In general, very very few connections
# should originate from this server subnet
iptables -A in-eth1 -j DROP
```

client

For the client network, we're going to focus on traffic bound for the external interface. Connections from the client network to the server or VPN interface will not be restricted at this time. If our IDS on the server subnet shows anything bad or we get complaints from our partners on the other end of the VPN tunnels, we may need to change this policy in the future and restrict it more.

```
# basic egress rule.  Drop any spoofed packets.
iptables -A in-eth2 -s ! 10.1.3.0/24 -j DROP
```

Allow only pre-defined protocols: ssh, http, https, pop3s, smtp, smtps, DNS queries, and ftp. The statefull features of our IPTables software will keep track of the connections, in particular the DNS (because it uses UDP) and ftp connections (because the server initiates connections to the client), are difficult to deal with when using a stateless packet filtering "firewall".

```
iptables –A in-eth2 –p TPC -d ! 10.0.0.0/8 --dport 22 –j ACCEPT
iptables –A in-eth2 –p TCP -d ! 10.0.0.0/8 --dport 80 –j ACCEPT
iptables –A in-eth2 –p TCP -d ! 10.0.0.0/8 --dport 443 –j ACCEPT
iptables –A in-eth2 –p TCP -d ! 10.0.0.0/8 --dport 995 –j ACCEPT
iptables –A in-eth2 –p TCP -d ! 10.0.0.0/8 --dport 25 –j ACCEPT
iptables –A in-eth2 –p TCP -d ! 10.0.0.0/8 --dport 465 –j ACCEPT
iptables –A in-eth2 –p UDP -d ! 10.0.0.0/8 --dport 53 –j ACCEPT
iptables –A in-eth2 –p TCP -d ! 10.0.0.0/8 --dport 21 –j ACCEPT
```

I use the REJECT flag here instead of the DROP. Because the traffic being dropped here is probably from a mostly friendly source. The REJECT will send a bit of ICMP to tell them we're blocking this REJECT on purpose

```
iptables -A in-eth2 -j
```

VPN

This chain can be used to restrict what people coming in through the VPN have access to.

```
# allow the Ravlin10 box to send it's syslog to the
# syslog server in the sever subnet
iptables -A in-eth3 -p UDP -s 10.1.1.1 -d 10.1.2.2 --dport 514 -j ACCEPT
```

allow traffic to the Oracle database and fileserver #1. Special effort has been made to put anything that needs to be shared over the VPN onto a single fileserver. Access to all other machines is blocked.

```
iptables -A in-eth3 -s 10.0.0.0/8 -d 10.1.2.5 -j ACCEPT
iptables -A in-eth3 -s 10.0.0.0/8 -d 10.1.2.6 -j ACCEPT
```

I use the REJECT flag here instead of the DROP. Because the traffic being dropped here is probably from a mostly friendly source. The REJECT will send a bit of ICMP to tell them we're blocking this on purpose.

```
iptables -A in-eth3 -j REJECT
```

- **VPN hardware (RedCreek Ravlin10 firmware v3.47)**

The following is a description of how the RedCreek Ravlin10 is configured at the GIAC corporate site. Configuration of the remote end of the VPN is similar, but I will not detail it here.

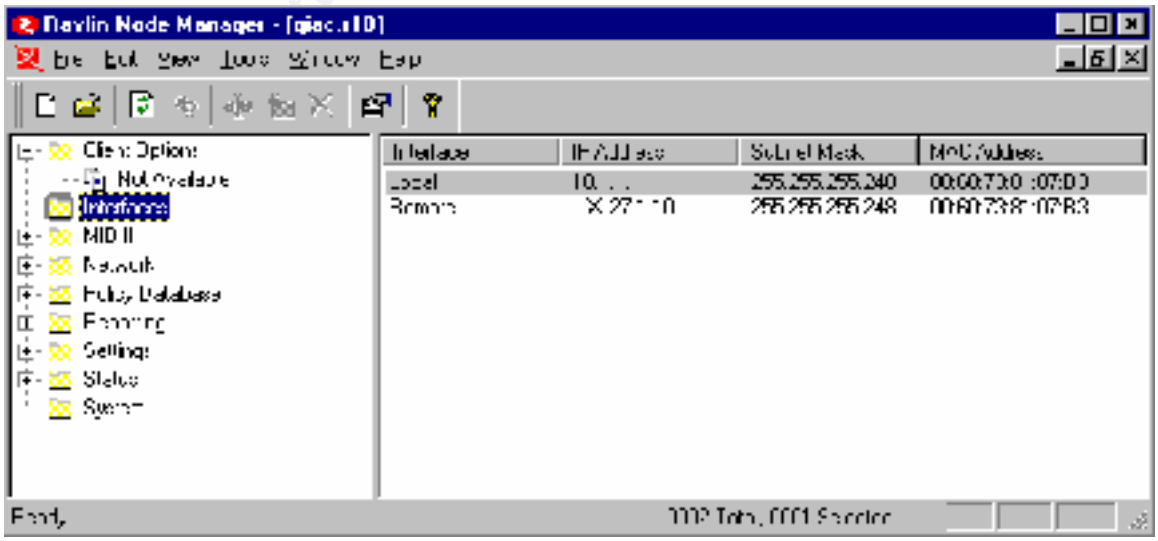
Ravlin VPN hardware is configured using a Windows GUI called "Ravlin Node Manager". That software uses SNMP along with a password to communicate and configure the device. Because SNMP is not a very secure protocol, the following have been done to reduce the risk as much as possible:

- SNMP traffic is not allowed on the external interface (blocked by firewall #1), so configuration can only be done from the internal network.
- The read and write SNMP community strings are treated like passwords and are changed regularly.
- The password is kept as secure as possible and also changed regularly.

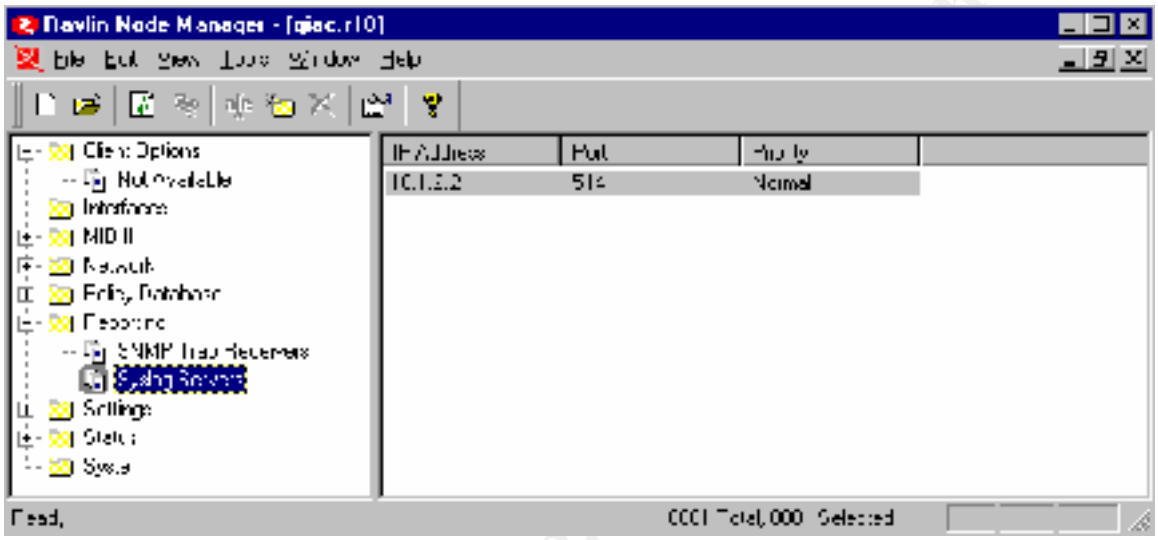
ISAKMP will be used for authentication. ESP tunnel mode will be used for the encapsulation. 3 key Triple DES will be used for the encryption. ESP was chosen over Authentication Header (AH) because of the better Data Integrity checking it algorithms that ESP provides for. A security ID number that is unique to each Ravlin box is used as the shared secret to establish the authentication.

After connecting to the hardware with the Ravlin Node Manager software and authenticating with the SNMP strings and the password, the following GUI screens are used to configure the device.

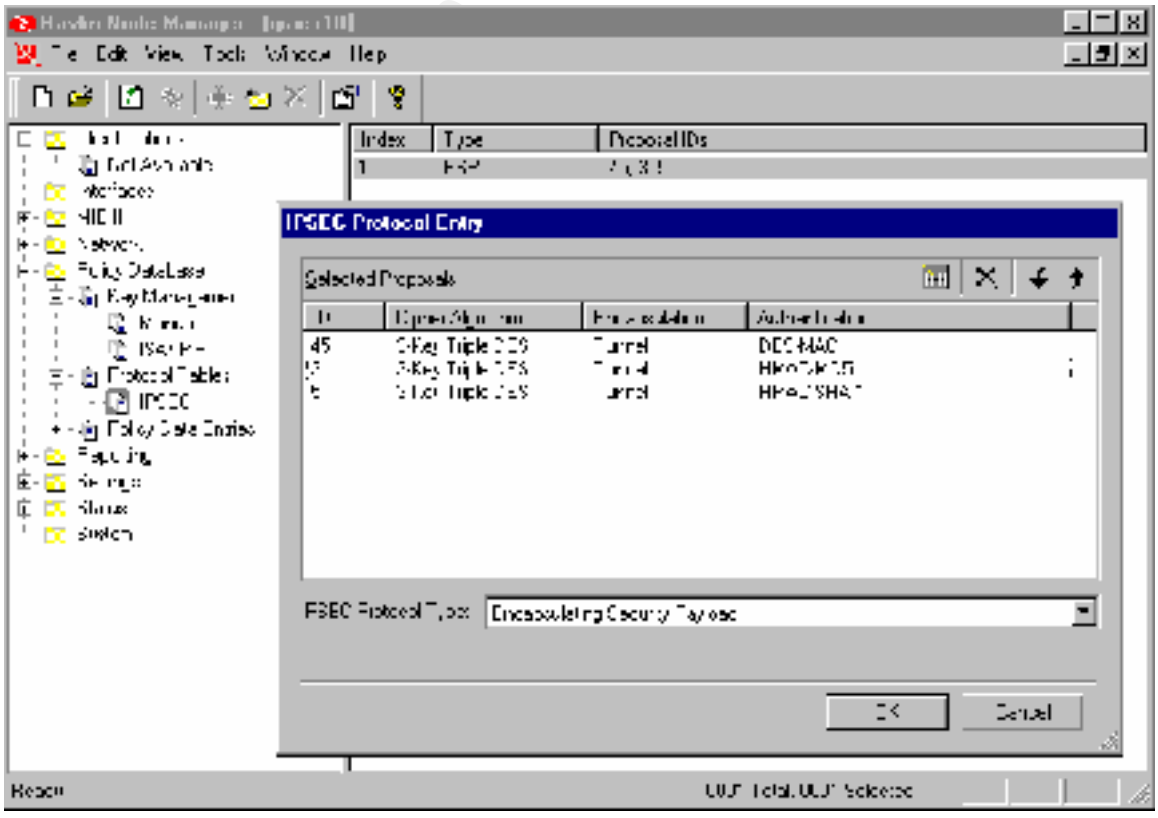
First, IP addresses and netmasks are assigned to the external and internal interfaces of the Ravlin.



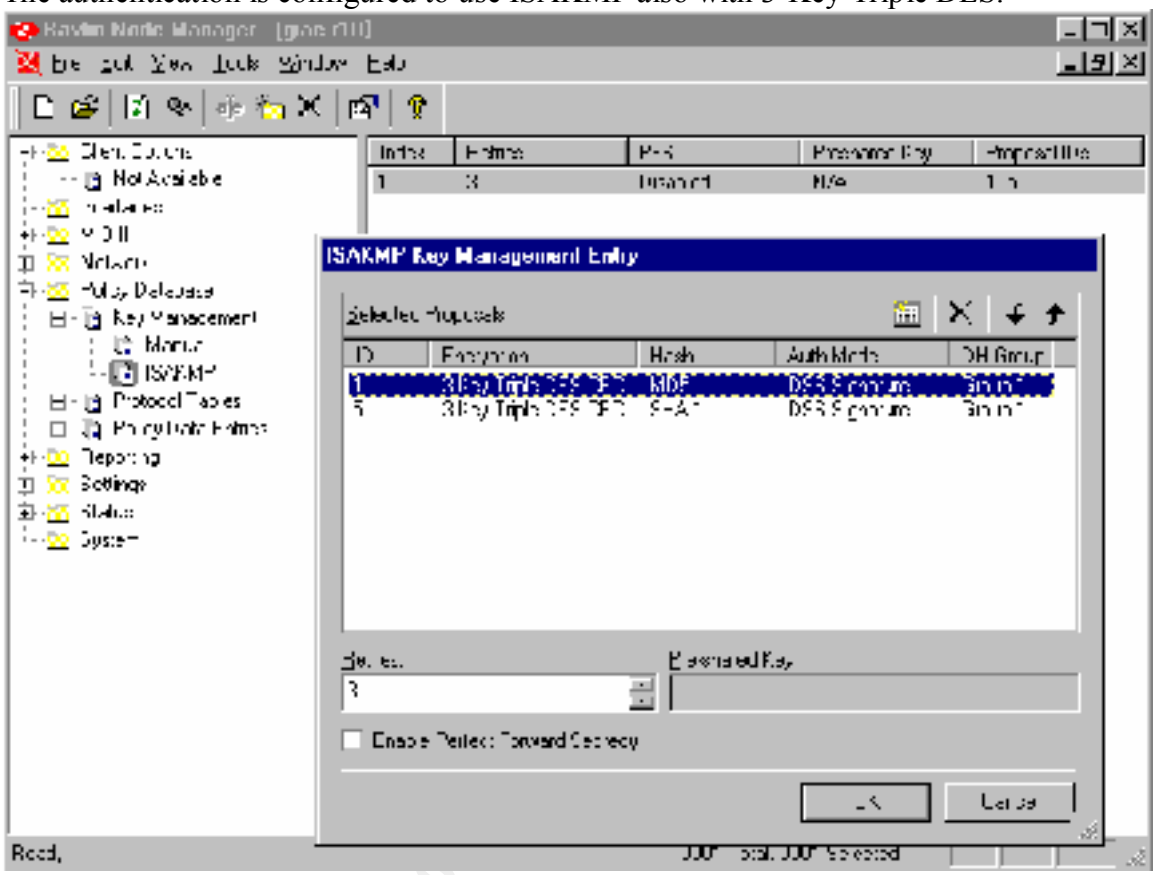
The syslog is configured to be sent to the syslog server in the internal server subnet. SNMP traps are turned off because GIAC does not currently have a NMS (network management system). If they implement a system like HP OpenView or CA Unicenter in the future, SNMP traps will be configured for those systems.



The protocol is configured to be ESP using 3-Key Triple DES tunneling mode.



The authentication is configured to use ISAKMP also with 3-Key Triple DES.



Assignment 3 – Audit Your Security Architecture

Assignment Text:

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

1. Plan the assessment. Describe the technical approach you recommend to assess your perimeter. Be certain to include considerations such as what shift or day you would do the assessment. Estimate costs and level of effort. Identify risks and considerations.
2. Implement the assessment. Validate that the Primary Firewall is actually implementing the security policy. Be certain to state exactly how you do this, including the tools and commands used. Include screen shots in your report if possible.
3. Conduct a perimeter analysis. Based on your assessment (and referring to data from your assessment), analyze the perimeter defense and make recommendations for improvements or alternate architectures. Diagrams are strongly recommended for this part of the assignment.

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

Solution:

Plan the assessment

As part of a comprehensive information systems audit of GIAC Enterprises, I will address the issues involved in the primary firewall from assignments 1 and 2. In my design I've labeled the primary firewall as Firewall #1. It's configuration consists of a Cisco Pix 515-UR version 5.3 with four ethernet interfaces.

Caveat: it should be noted that because this paper addresses a security architecture design that is still not implemented, there will not be as much real world output and screen shots as I would like there to be.

Scope: This audit will address the configuration and integrity of the primary firewall only. It will not address other component of the security architecture like the border router, secondary firewall, or VPN hardware. Additionally, system level security of the nodes which are being protected by the primary firewall will not be addressed here. System level security is complex and more difficult than most people realize. A description of proper system level security could easily fill an entire paper itself (ie. GCNT & GCUX) I will also not be addressing the security of the services provided in

the service network. Each service has it's own security issues, whether it's DNS, IIS, apache, sshd, or the infamous BIND.

Goals: The goals of this audit will be twofold. Firstly, we will validate that the primary firewall is actually implementing the security policy intended. Secondly, we will attempt to discover any vulnerabilities which were not even addressed by the implemented security policy.

Tools: The primary job of a non-proxying firewall is to act as an intelligent router, inspecting packets' IP protocol type, source address, destination address, and relevant ports. To test that the firewall is performing this task properly, we will want to send some test packets to the firewall to see if they are passed or dropped as expected. nmap and hping2 are two excellent tools for generating specific known packets. To verify that the packets do or do not pass the firewall, we'll use a combination of listening tools and also watch the firewall's own logs. The logs should indicate which packets are being dropped and which rule caused that drop. The listening tool that we will use is called ethereal. Ethereal is a graphical front end to tcpdump, a popular unix packet sniffing tool.

Schedule: The audit will be performed during the day shift on a weekday. This will insure the availability of important IT personnel for assistance and troubleshooting. There should be no disruptions in service created by performing the audit, so it does not need to be done off-shift. If any problems are discovered, they will be noted, processed through change-control and implemented after hours according to the corporate change-control policy. Time required for the audit is approximately 8 hours for planning, 8 hours for implementation, and 8 hours for analysis and documentation. On-site, salaried, full time staff members of GIAC Enterprises will be used to conduct the analysis, so the cost will be approximately $1200 in fractional salary. It would also be beneficial to hire a third party to perform an independent security audit. Their costs would likely be $200-$400 per hour and is beyond the scope of this paper.

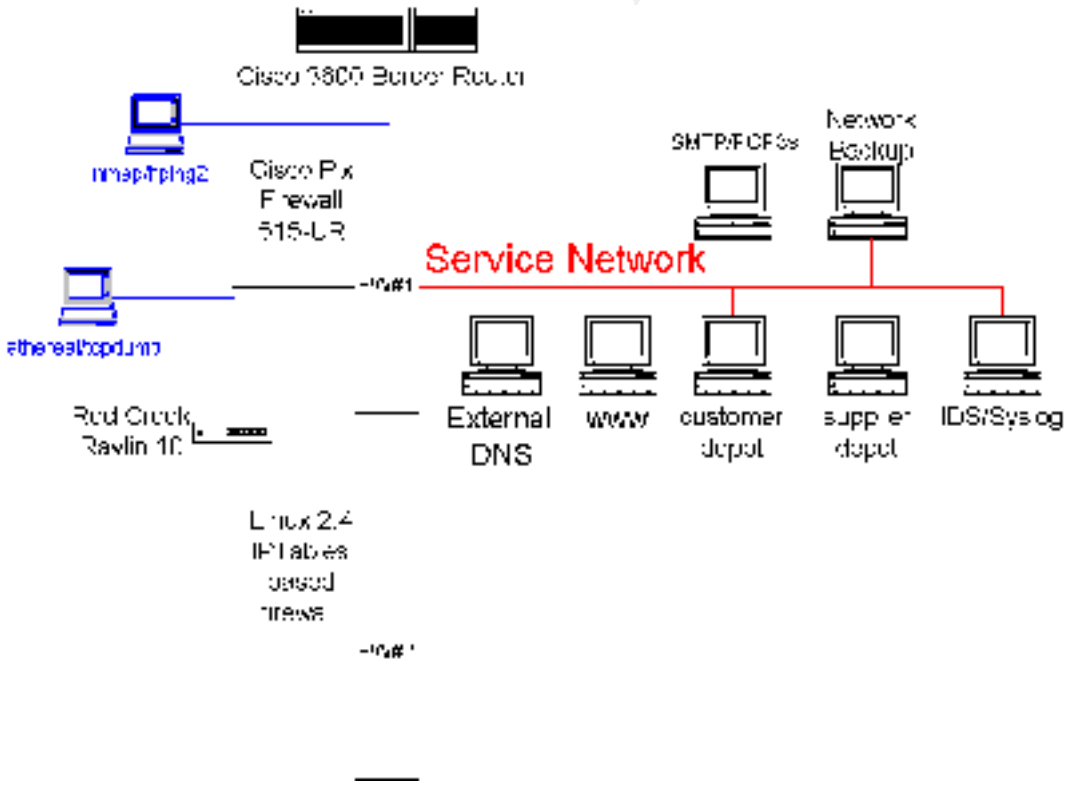
What to check: There are four network interfaces on this firewall and we will want to check each combination of source and destination. The testing matrix for this assessment will be fairly large.

Originating Interface	Destination Interface	Notes
Outside	Service Network	
Outside	VPN	
Outside	Inside	
Service Network	Outside	
Service Network	VPN	
Service Network	Inside	
VPN	Outside	
VPN	Service Network	
VPN	Inside	
Inside	Outside	
Inside	VPN	
Inside	Service Network	

Implement the assessment

Before testing begins, we should have already familiarized ourselves with normal traffic patterns. Logfiles from the firewall, and ethereal captures should be made at various times during the day throughout an entire month. This will ensure that we are comfortable looking at normal traffic patterns on our network and can more easily recognize abnormalities.

Setup: Two unix machines will be used for the assessment. One will be loaded with nmap and hping2. The other will be loaded with tcpdump and ethereal. They will be connected to the network switches on either side of the firewall. The switch that the detection box (tcpdump and ethereal) is connected to will be configured to mirror all traffic to that port. That way the detection box will be able to listen to all of the traffic coming out of that side of the firewall. All subnet masks in this design were purposely set at 29 bits or fewer to facilitate this sort of testing. If a subnet mask of 30 bits were to be used, there would be no IP available to insert probes like this. The following diagram illustrates the setup for testing access from the outside interface to the VPN interface of Firewall #1.



The following test should be performed for each line of our testing matrix above.

1. ping the firewall (ping the interface to which you are connected).

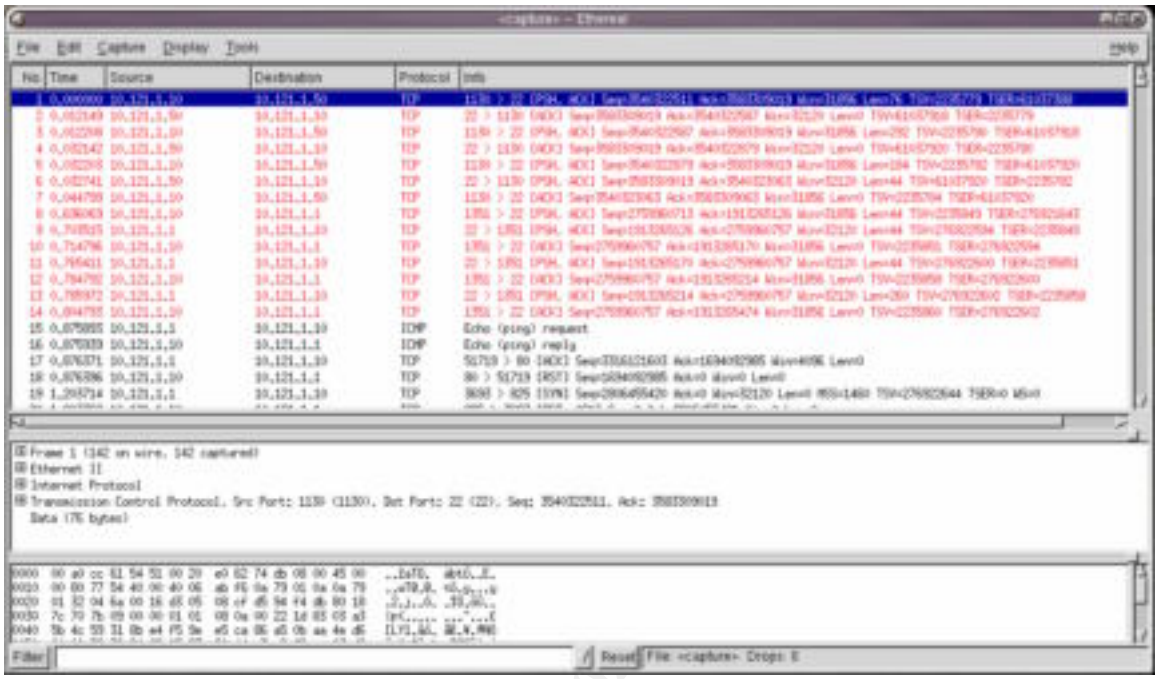
2. ping each IP address off of the downstream interface. In the example above, this would be X.27.1.9-X.27.1.14. For the service network, this would include a ping of each node on the network.
3. do a TCP nmap scan (nmap –sT) of the firewall
4. do a TCP nmap scan (nmap –sT) of each IP address off of the downstream interface (like #2)
5. do a UDP nmap scan (nmap –sU) of the firewall
6. do a UDP nmap scan (nmap –sU) of each IP address off of the downstream interface
7. do a SYN-FIN hping2 scan (hping2 –S –F -p <destination port>) of the firewall
8. do a SYN-FIN hping2 scan (hping2 –S –F -p <destination port>) of each IP address off of the downstream interface

In each case, the output from the ethereal/tcpdump machine should be examined to discover any improper packets that are being allowed through. The output of nmap should also be examined to verify that the ports it shows as open are the only ones we want open. We should also examine the firewall's logfiles to verify that it is logging the dropped packets as expected.

Sample nmap output:

```
[root@failte /root]# nmap 10.121.1.10

Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
Unable to find nmap-services!  Resorting to /etc/services
Interesting ports on craic.chriso.pnn.com (10.121.1.10):
(The 1061 ports scanned but not shown below are in state: closed)
Port       State       Service
22/tcp     open        ssh

Nmap run completed -- 1 IP address (1 host up) scanned in 2 seconds
```

Sample ethereal screenshot:



In order to check out egress filters, the following test should be performed on each interface of the firewall.

- Try to send a TCP SYN packet to a node on the downstream interface using a spoofed IP address. This can be done by using hping2 (hping2 –S –a <spoofed IP address> <destination>).

Of course, a final test (or initial test, depending upon your point in the implementation) would be to verify that each service we're providing is actually functioning. This means we should check that our VPN tunnel to our partners is functioning . We should check that each server in the service network that should be reachable is functioning as expected.

Although it is beyond the scope of this paper, these test are also an excellent time to verify that your IDS systems are functioning as you would like them to. IDS systems should be watching your important network segments for suspicious activity like nmap portscans and spoofed IP addresses.

Conduct a perimeter analysis

The results of the security assessment that was conducted in the previous step will surely show several things.

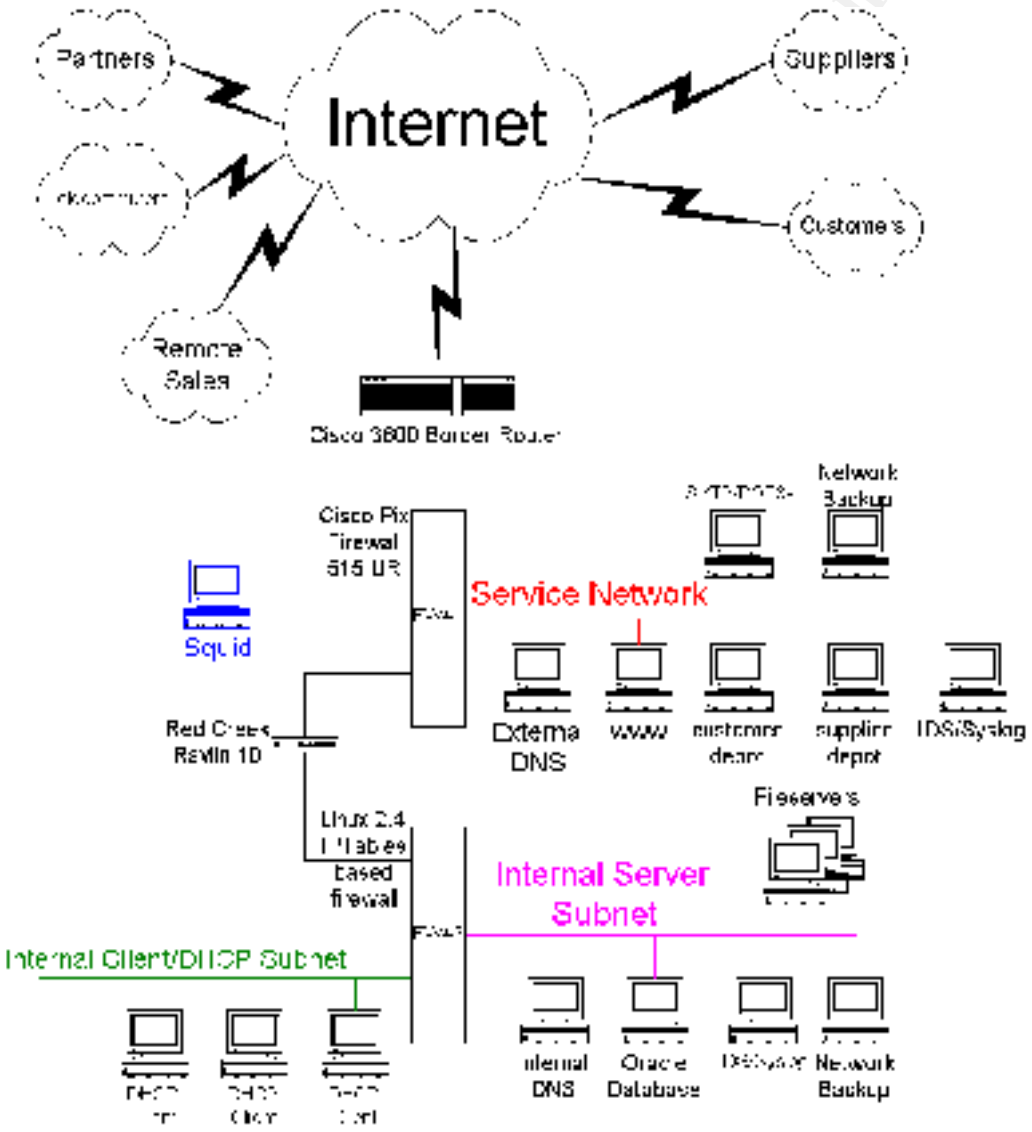
1. It will show the errors we have made in implementing our chosen security policy. These errors might be syntax errors and accidentally omitted rules.
2. It will verify that the holes that we knew we were leaving open, are indeed, open.

Anything that falls under #1 above, should be corrected as soon as possible.

Anything that falls under #2 should be reviewed by all parties involved to verify that the risks that are being allowed are acceptable. This may result in changes in the design or a decision to keep things the way they are.

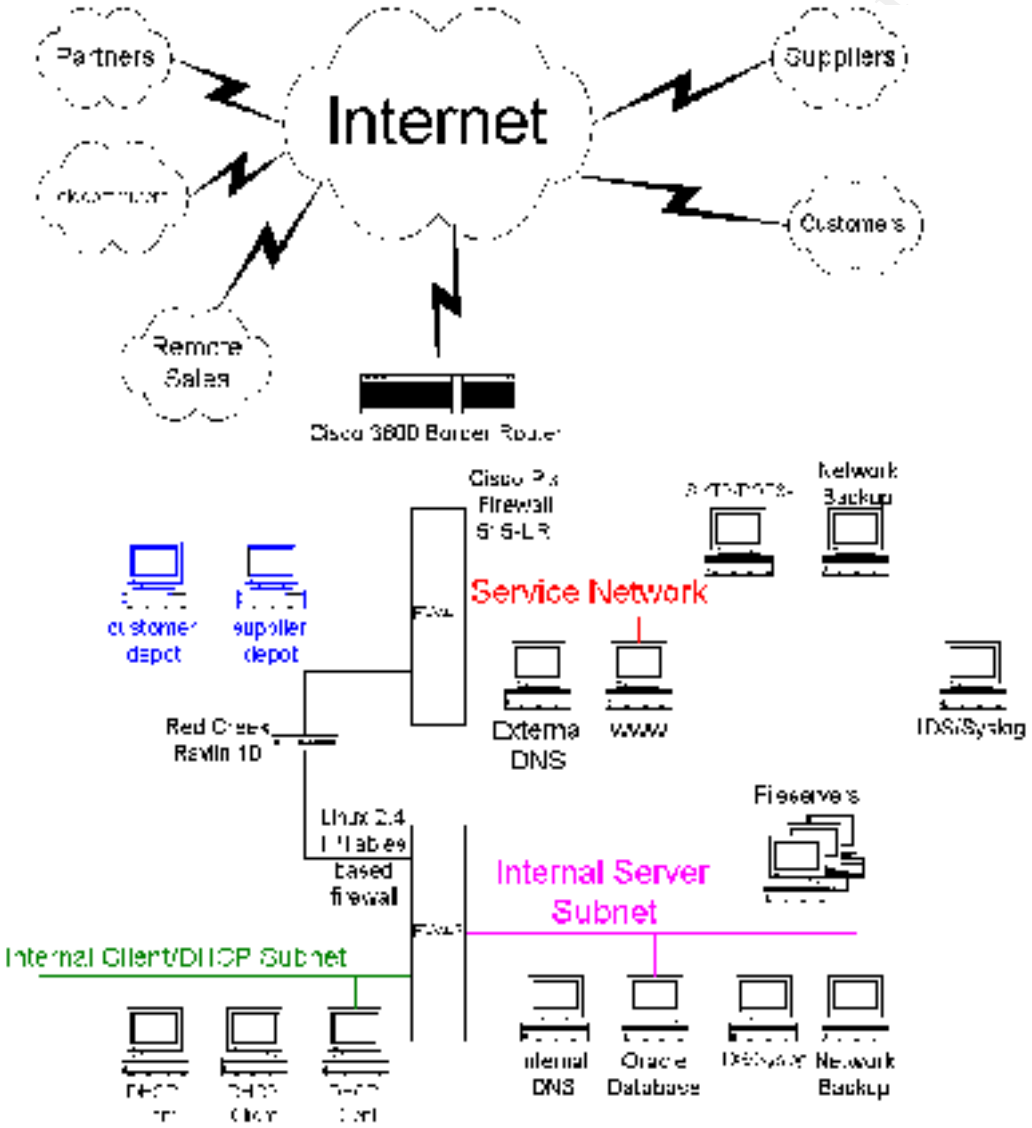
When considering alternate security designs, there are as many different designs as there are security engineers in the world. I've suggested two alternates below that are small variations on my original design. There are, of course, far more radical changes that are possible as well.

Add a proxy.
An application proxy or proxying firewall can provide additional security at the expense of some performance and complexity. I've illustrated a variation below that includes a squid web proxy server on it's own dedicated firewall interface. This squid proxy could be used for outgoing web proxying for GIAC office workers, but this placement is actually better suited to act as a reverse web proxy server for the public web server on the service network, or even the depot servers, since they are web based also.



Move the depots to their own segment

Another design alternative might involve the separation of the depot servers from the publicly available servers on the service network. The depot servers are only meant to be access by specific customers and suppliers. They contain data that is core to the business and should not be accessed by the general public. This difference in classification and access patterns suggests that they would be more secure if isolated from the public servers.



In the end, a list of known and verified vulnerabilities should be formally acknowledged and signed by the security engineer and management. Further security audit should be conducted on a regular basis.

Assignment 4 – Design Under Fire

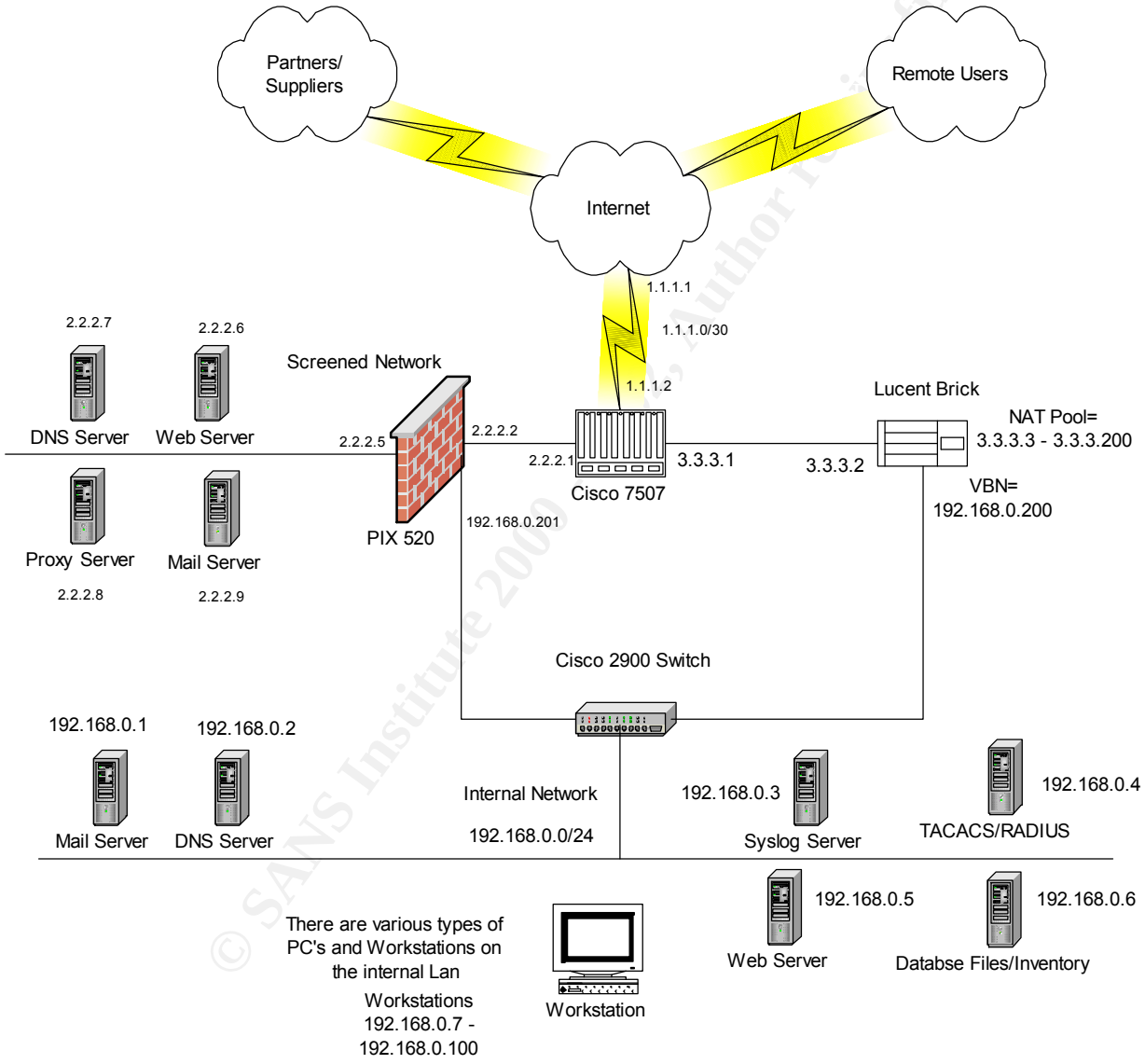
Assignment text:
The purpose of this exercise is to help you think about threats to your network and therefore develop a more robust design. Keep in mind that the next certification group will be attacking your architecture!

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

1. An attack against the firewall itself. Research vulnerabilities that have been found for the type of firewall chosen for the design. Chose an attack and explain the results of running that attack against the firewall.
2. A denial of service attack. Subject the design to a theoretical attack from 50 compromised cable modem/DSL systems using TCP SYN, UDP, or ICMP floods. Describe the countermeasures that can be put into place to mitigate the attack that you chose.
3. An attack plan to compromise an internal system through the perimeter system. Select a target, explain your reasons for choosing that target, and describe the process to compromise the target.

Solution:

The design that I've chosen to use for this section of the paper is that of Larry Koons (http://www.sans.org/y2k/practical/Larry_Koons_GCFW.doc) Larry's design uses a Cisco Pix firewall running version 5.2.3 of the pix software.



Attack the firewall itself.

By using the SecurityFocus vulnerability database, I was able to discover a flaw in the Pix handling of SMTP traffic. Bugtraq ID 1698 entitled "Cisco PIX Firewall SMTP Content Filtering EvasionVulnerability" The description from bugtraq is the following:

"*Like other firewalls, the Cisco PIX Firewall implements technology that reads the contents of packets passing through it for application-level filtering. In the case of SMTP, it can be configured so only certain smtp commands can be allowed through (for example, dropping extra functionality, such as HELP or commands that could be a security concern, like EXPN or VRFY). When recieving messages, it allows all text through between "data" and "<CR><LF><CR><LF>.<CR><LF>", as this is where the body of the message would normally go and there could be words in it that are smtp commands which shouldn't be filtered. Due to the nature of SMTP and flaws in exceptional condition handling of PIX, it is reportedly possible to evade the smtp command restrictions by tricking the firewall into thinking the body of the message is being sent when it isn't.*

During communication with an smtp server, if the "data" command is sent before the more important information is sent, such as "rcpt to", the smtp server will return error 503, saying that rcpt was required. The firewall, however, thinks everything is alright and will let everything through until recieving "<CR><LF><CR><LF>.<CR><LF>". It is then possible for the attacker to do whatever he wishes on the email server."

Larry's design is vulnerable to this attack because the Pix is the only firewall between the mail server and the internet. Using this exploit against the firewall/mail server in this case, would possibly result in the ability to run VRFY SMTP commands, thereby discovering the usernames of GIAC Enterprise employees. These usernames would then be powerful ammunition to launch further attacks, or even to sell to spammers.

A distributed denial of service (DDOS) attack.

The TFN "Tribal Flood Network" DDOS tool would seem most appropriate for this job. TFN clients can be directed to participate in a DDOS attack using TCP SYN, UDP, or ICMP packets directed at a particular target. In this case, the target will be Larry's Cisco Pix firewall. This particular attack, if deployed properly, will render the Pix firewall unresponsive for the duration of the attack

The solution to this problem is to use the rate-limit command now provided with the Pix OS. The rate-limit command can be used to direct a maximum SYN connection rate within a particular ACL. When that rate is exceeded, the packets should be dropped. The specific rate used is a matter of some experimentation, but 30% is a good starting point.

An internal system compromise.

These days the easy targets all seem to be web servers. Since he doesn't specify, I'll assume Larry's web server on the screened network is running Microsoft's IIS. The IIS web server is both very popular and full of security holes. I chose to use a buffer overflow attack on this IIS server.

NT IIS4 Buffer Overflow Vulnerability
Bugtraq ID 307

From the bugtraq description:
"A buffer overflow vulnerability in the way IIS handles requests within .HTM extensions allows remote attackers to execute arbitrary code on the target machine.

IIS supports a number of file extensions that require futher processing (i.e. .ASP, .IDC, .HTR). Whe a request is made for one of this file types a specific DLL processes it. A stack buffer overflow vulnerability exists in ISM.DLL while handling .HTR, .STM or .IDC extensions. The ISM.DLL filter is installed by default with IIS."

Buffer overflow exploits are particularly nasty. That phrase "execute arbitrary code" in the description means that an intruder using this exploit, can run whatever they feel like on this server. This would most likely be installing a backdoor into the system so that further attacks could be made.

The solution to this attack is simply brining IIS up to the proper patch level. The proper level in this case is NT 4.0 SP6.

Bibliography

Frank Keeney's cisco ACL config guide
http://pasadena.net/cisco/secure.html

Configuration Guide for the Cisco Secure PIX Firewall Version 5.3
http://cisco.com/univercd/cc/td/doc/product/iaabu/pix/pix_v53/index.htm

Linux 2.4 Packet Filtering HOWTO
http://netfilter.kernelnotes.org/unreliable-guides/packet-filtering-HOWTO/

Linux IPCHAINS-HOWTO
http://www.linuxdoc.org/HOWTO/IPCHAINS-HOWTO.html

Linux 2.4 NAT HOWTO
http://netfilter.kernelnotes.org/unreliable-guides/NAT-HOWTO/

RedCreek Ravlin10 web site
http://www.redcreek.com/products/ravlin10.html

Many man pages, including ethereal, tcpdump, nmap, hping2, iptables, ip, ifconfig, snort, and ipchains.

SecurityFocus vulnerability database
http://www.securityfocus.com/

Cisco's DDOS whitepaper
http://www.cisco.com/warp/public/707/newsflash.html

University of Washington's paper on The "Tribe Flood Network" distributed denial of service attack tool
http://staff.washington.edu/dittrich/misc/tfn.analysis

Inspiration, good ideas, and good examples were gleaned from many other GCFW practicals.