Global Information Assurance Certification Paper

Rune Magic, Changing the fabric of Security

GIAC Certified Firewall Analyst
Practical Assignment
Version 2.0

Marc Panet-Raymond

March 1, 2004

Table of contents

Executive Overview

The following paper is submitted to meet the requirements for GCFW analyst by designing, documenting and testing a network for Rune Enterprises[1]. Rune Enterprises is a small family business specializing in providing proverbs enhanced by Rune magic. Based on over whelming requests from its customers Rune Enterprises has expanded into e-business. Its first e-business venture was primarily out-sourced to their Internet Service Provider (ISP). As their e-business grew, management decided to develop some of the services themselves. Since Rune Enterprises has limited staff and expertise it has decided to work with a security consultant to assist in the network design.

The security consultant worked with Rune's technical support person, a family member, to complete the network architecture for Rune Enterprises. The consultants report includes a detailed security architecture; including the border router policy, primary firewall policy and the Virtual Private Network (VPN) policy. It also includes a tutorial on Rune Enterprises primary firewall to assist in the technology transfer to the support staff. To ensure the design will perform, as developed, an audit was planned, executed and analyzed for Rune Enterprises.

To demonstrate that security is a balance of managing risk and business objectives an analysis of a fictional competitor was also submitted to management. This attack analysis is to assist the support staff when talking with Rune management highlighting that network security is an ongoing process. Network security is not install and forget.

Security Architecture

Introduction

Rune Enterprises is a family run business specializing in proverbs enhanced by Rune magic. The business has expanded as the primary customer, by word of mouth, kept referring their contacts to Rune Enterprises. The owner of Rune Enterprises has until this point held off getting into e-business, preferring the traditional method of face to face business. With encouragement from their first large customer Rune Enterprises decided to launch into e-business. Since Rune Enterprises is a small family run business and did not have the expertise, at that time, it outsourced their first venture into e-business to their local ISP. The ISP provided all the technical expertise for their setup and proverb application. Since initially there was only a small amount of traffic the ISP used a commercial cable internet connection at Rune's site with a Linksys BEFSR41 router protecting the web server. The ISP provided all the network services, email, domain name system, (DNS), and Web site management. As business developed, and a

family member developed expertise, Rune Enterprise management decided to take control of their information technology. The plan was to bring the infrastructure in house with the exception of the DNS servers.

As business continued to grow well in their local area, management decided to look outside the local area. Therefore in conjunction with bringing the "network home" they started promoting their products over a greater geographical area with two dedicated traveling sales people.

Since it is still, essentially, a family run business with limited amount of resources, management decided to hire a security consultant to assist their technical support person in the design of the new network. This was based on a recommendation from a family acquaintance, who attended a SANS seminar explaining the need for defense in depth for their e-business.

The architecture designed in this paper is Rune Enterprises second venture into e-business.

Business drivers and access requirements

Rune Enterprises' primary business is enhancing fortunes received from suppliers by adding Sartan Rune magic[2]. Their secondary business is providing translation services, French and English, to its partner company, Patryn International. Patryn International is also used to translate it fortunes into Spanish, German, Chinese and other language requested by its corporate buyers.

Family members take proverbs purchased from their suppliers and enhance them using magic passed on from generation to generation. This process is done on the "Secure application server." Access to this server is limited from two executives desktop systems only. Once the new proverbs have been enhanced a nightly secure shell (`ssh`)[3] copy moves them to the production application server. Here they are packaged and priced. These packaged proverbs are then copied by a nightly `ssh` copy job to the web server.

Thus, the crown jewels for Rune Enterprises would be their secure application server since an attacker could possibility reverse engineer the enhanced proverbs and learn trade secrets. The secure application server was placed on the management segment for an additional layer of protection. Since the current technical support person is related to the family, insider risk is minimized.

As part of their analysis Rune Enterprises examined how each of the following groups would interact with their network: their customers, suppliers, partners, their traveling sales people and their staff.

Rune Customers

This group has already purchased at least one proverb from Rune. This includes both corporate and individuals. Corporate customers usually purchase large volumes of proverbs to subsequently use in promotional type material. Corporate customers have already set up financial arrangements with Rune enterprises

Individual customers usually purchase smaller volumes of proverbs and pay with credit cards.

General web site browsing uses the `http` protocol. Once the customer has decided to purchase they then use the `https` protocol to protect their information.

Access is directed to Rune's public proverb application running on the web site, `www.rune.ca`. The web site requires a web browser supporting `http` and `https`. Email can also be sent through the web site or by sending mail to `info@rune.ca`. Email is directed to, through an `mx` record to the public mail server `mail.rune.ca`

Rune Suppliers

This group provides Rune Enterprises with the raw proverb material. This group sometimes changes based on the current customer trends for proverbs. Suppliers are granted access to a small part of the production application server through a VPN peer connection. The VPN peer connection limits access to the Production application server only. This allows them to bulk copy proverbs to a pre-production database that runs on the production application server.

Access requirements are a web browser supporting `http` and `https`. `ssh/scp` is used for bulk uploads through the web application. Access is via the VPN peer connection.

Rune Partner

This group provides Rune with translation and reselling services. This business has made a long term contractual commitment with Rune Enterprises and has established more direct network connection to Rune's network. This group does not change often. Part of the agreement to connect the networks together was that each company have appropriate protection in place including a hardware firewall device. The partner provides translation services for all languages except French and English. Rune Enterprises provides their own translation services from English to

French.

Access requirements are a web browser supporting `http` and `https`. `ssh/scp` is used for bulk transfers through the web application. Email between the companies still travels through the internet. Pretty Good Privacy (pgp/gpg) is used if required.

Rune Executive staff

This group is mainly Rune family members and their close relatives. At times this group may travel and may require remote access. Additional steps have been taken to ensure their laptops are secured during travel, for example encrypted file systems and additional passwords. This group has been provided with additional security awareness training.

Access is provided through VPN remote access. Access is granted to both the production application server and main server primarily to the accounting and operations systems. Access to the secure application server is not granted through remote access only local access is allowed.

The overall internet access policy states that access is limited to business use only. Working with both management and technical support they have determined that employees will have internet access with the following protocols: `http, https, smtp, dns(udp). ftp` is not allowed since management does not want to risk copyright infringement by its employees who download licensed materials. All other protocols have been denied. If another protocol is required sufficient business justification is required. The technical support staff has in addition, `ssh` access.

Rune Mobile sales

This group are the two traveling sales people that primarily use remote access. Since these laptops could provide direct access to the internal network each sales person has also been provided with additional training and awareness is overall security.

Each sales person has their own password to access the VPN since they are not numerous. This approach would change if they increase the number of sales people and the business continues to grow. Lightweight Directory Access protocol (LDAP) may be investigated if required. The VPN client is configured with no split horizon, that is when connected to the internal network their other network connection is disabled.

Access is required to the sales system on the production application server and main server.

Rune IT Staff

This group includes the technical support person, the two programmers and consultants, when required. This group supports the network infrastructure for Rune Enterprises. At times support may be required from off site locations when no local staff is onsite.

Additional steps have been taken to ensure the support laptops are secured during travel, for example encrypted file systems and additional passwords. This group has also been provided with additional security awareness training

Access requirements to all systems is via the VPN.

Rune Business staff

This group is the remainder of the Rune Enterprise administrative staff. This group only requires onsite access and has no requirement for remote access.

The general public

This group is the group that is not any of the above group. This group is viewed as potential customers. This may be through referral web sites and traditional advertising.

Access requirements are a web browser supporting `http` and `https`.

Network design and components

The proposed design is based on the “Defense in Depth” principle, that more than one component provides overall security for Rune Enterprises. This design incorporates more than one device providing protection including various manufacturers, architectures and platforms. This ensures that a vulnerability or weakness in one device does not completely eliminate the security posture. In addition, defense in depth, potentially could slow down an attacker to allow support staff to handle the incident appropriately. The components selected also had to meet both technical and financial objectives.

Internet access at Rune Enterprises headquarters location is limited to two providers, the telephone or cable company. After a cost benefit analysis it was determined that a basic commercial account with the cable provider would be the best fit at this time. This account provided for one static IP address, sanitized

address, 2.2.2.1, and sufficient overall transfer volume without penalty.

Working with Rune management the following requirements were developed.

Technical requirements:

- Secure, providing defense in depth
- Minimal installation duration
- Minimal complexity for support
- Minimal support requirements
- Interoperate with exiting administrative systems
- Allow for future growth

Financial requirements:

- Respect the capital budget
- Require no additional staff for technical support
- Leverage existing equipment
- Minimize license costs

This design also allowed for the migration of the existing, non internet connected, administrative Windows server and workstations.

With the above objectives set the following security components were selected:

- Border router – Cisco 1720 router with Ethernet module
- Primary firewall – Cisco PIX 501
- VPN – Cisco PIX 501
- Secondary firewalls – `pf` packet filter on OpenBSD version 3.4

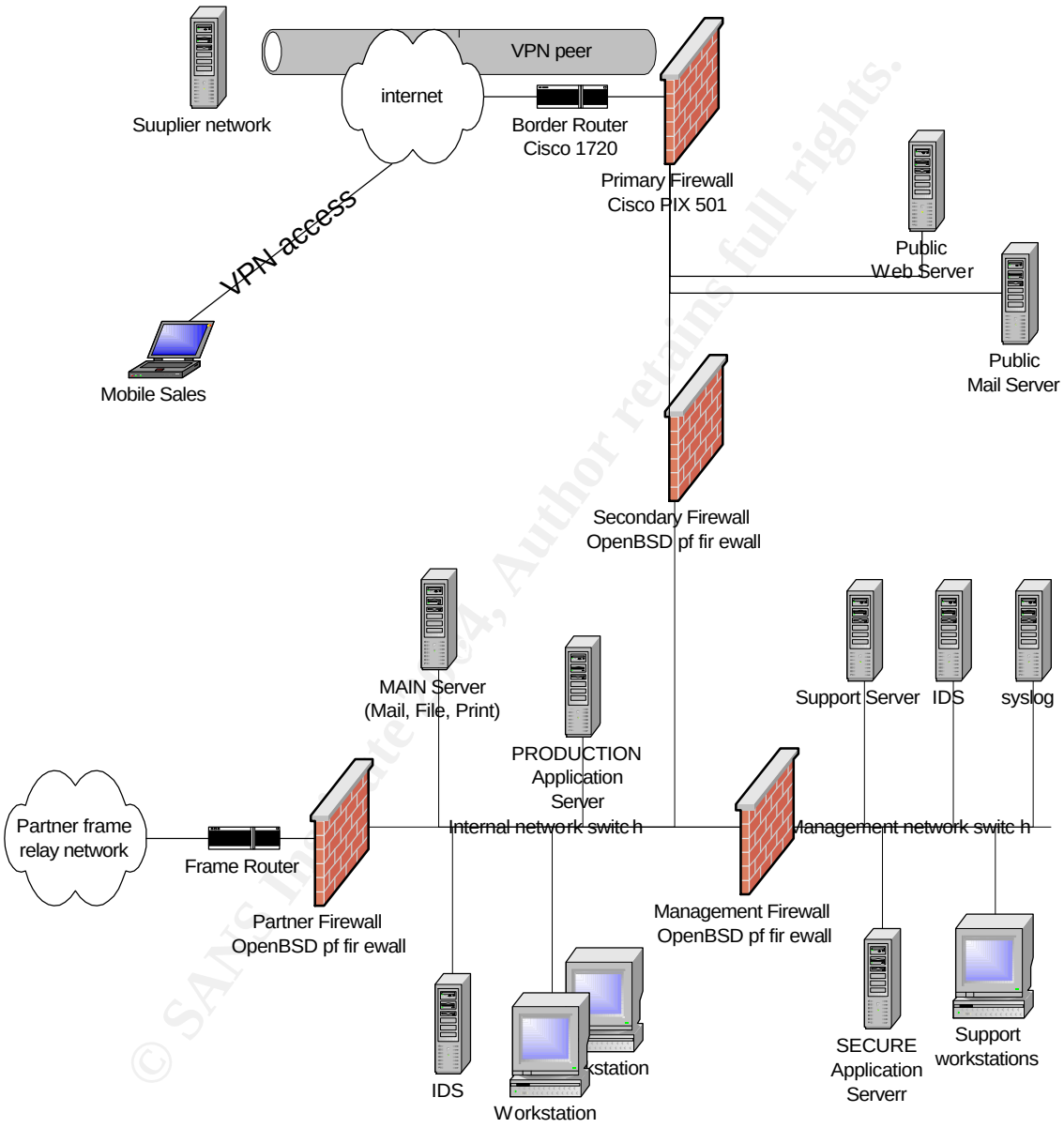
Various other security architectures and designs were examined and either for budgetary or financial reasons were not selected for the design at this time.

- Standalone VPN device
- Web Proxy Server
- Fail over design
- Separate server segment
- Separate workstation segment
- Separate syslog segment
- Separate IDS network

The following areas would be addressed in the overall infrastructure but are outside the scope of this report.

- Physical security
- Systems backups
- Patch management
- Tripwire and host hardening
- Company policies and procedures.
- Windows system protection

Rune Enterprises Network



As described in the IP addressing section, following, each specific area, subnet, has its own IP address range. In addition, the addressing of the hosts was not sequential to assist in the detection of scanning activities. Since the addresses are not sequential any attempt to connect to sequential hosts could be viewed as a reconnaissance attempt.

Cisco was chosen for the border router since it would allow Rune Enterprises to leverage knowledge gained on support to be used on larger equipment as the company grows it network. Support IOS on a 1720 is very similar to 2600 or 3700 class routers.

The primary firewall is a Cisco PIX 501 running firewall software version 6.3(1). Since the device target market is the small office home offices (SOHO) it only has two interfaces: an outside and an inside interface. The PIX also provided Rune Enterprise with a VPN solution not requiring an additional device to support. The PIX supports both VPN remote access and VPN peer connections. The PIX 501 does not support Virtual Local Area Networks (VLAN). This was not a hindrance since the objective was to keep the design straightforward and reduce complexity. VLANs add a certain amount of complexity that was deemed not required at this time. A requirement that would have been beneficial would be the ability to create a De-militarized Zone (DMZ) on the firewall. This feature is not available on the PIX 501 or 506. To receive this feature you would have to purchase the PIX 515E which at this time did not meet the budget restrictions. A 50 user license was chosen based on the anticipated volume of internet traffic.

The PIX 501 can handle up to 7500 simultaneous connections and up to 10 VPN peers. This performance would able to handle the expected traffic of 10 to 100 sales per hour.

Cisco was also chosen for the primary firewall because of financial reasons. It was shown to management as a cost savings to purchase firewall devices that also included VPN support. Another argument convincing management was the knowledge gained on the PIX 501 can be used to support the larger PIX systems when the time comes to upgrade the network.

The web and mail server are connected directly into two of the four switched ports of the PIX. The mail server acts as a proxy sending email to the internal exchange server. It also accepts internal email for delivery to the internet. Since we are using port forwarding we can only have one server supplying the service on the well known ports. Thus, we cannot do any load balancing; our volume estimates do not anticipate needing this functionality.

DNS, as mentioned above, is provided by the local ISP, sanitized address 2.2.0.8 and 2.2.0.10. The main reason is that Rune Enterprises only has two hostnames advertised, `www.rune.ca` and `mail.rune.ca`, both pointing to the single public IP address, supporting the web site and mail server respectively.

To continue our defense in depth approach another vender and platform was

selected for the secondary firewall. OpenBSD with `pf` packet filtering was chosen for the following reasons. Its code base has been audited for security completeness[4] and based on a report[5] it requires a more knowledgeable attacker to compromise. In addition, it is good to support another small Canadian startup company.

Patryn International, the partner company, is connected using a frame relay connection, based on historical reasons. The frame relay router is managed by the leased line provider. Thus Rune Enterprises has no control over the frame relay router. As a result of this another secondary firewall is added between the partner and the internal network.

Behind the secondary firewall is where most of the business for Rune Enterprises takes place. He we will find a mostly Microsoft environment (Windows 2000 server and XP professional) supported by the technical support person. In addition, two programmers provide most of the application support from this environment.

Each subnet is watched by a snort[6], version 2.1.0 Intrusion Detection System (IDS). Each IDS is based on a Redhat Linux version 9.0 hardened based on SANS step-by-step[7] [8]. Each IDS is connected to the switch's spanning port to ensure it sees all the traffic on the network. At this point in time the IDS's have addressable IP address. In the future it would be advantageous to each IDS listen without an IP address and communicate through a second interface to the central IDS system.

The IT support staff and systems are behind another OpenBSD secondary firewall. Here are located the syslog, IDS management, and support servers. In addition, the secure application server is located here for security reasons. Only a limited number of people have access to the Secure application server that contain the Rune enhanced proverbs.

Public Servers

The public web server has the following configuration.

- Compaq Proliant 1600
- Linux Redhat 9.0 hardened and fully patched
- Apache 2.0.48
- Tripwire, version 2.3.1-14
- Openssh, version 3.7.1p1
- Openssl, version 0.9.6l
- mysql, version 4.0
- iptables

The public email server has the following configuration.

- Compaq Proliant 1600
- Linux Redhat 9.0 hardened and fully patched
- postfix version 2.0.18
- Tripwire, version 2.3.1-14
- Openssh, version 3.7.1p1
- iptables

The only local accounts on the public servers are for the technical support only. Also, all unused applications and services have been removed. syslogs messages are forwarded to the central syslog server.

Main Server

The Main server has the following configuration.

- Compaq Proliant 1600
- Windows 2000 Server – SP4, patched
- Exchange 2000 – SP3, patched

Production Application Server

The Production application server has the following configuration.

- Compaq Proliant 1600
- Linux Redhat 9.0 hardened and fully patched
- Apache 2.0.48
- Tripwire, version 2.3.1-14
- Openssh, version 3.7.1p1
- Openssl, version 0.9.6l
- mysql, version 4.0
- iptables

At this time there is no business requirement for any wireless networking. Also, the management perceives this to be complex and not totally secure.

IP addressing scheme

The IP addressing strategy is approach in two sections: the pubic and the private addresses. Since our ISP commercial service only supplies us with one public IP address the public section is rather basic. As Rune Enterprises grows we will have to investigate other options. The private addressing is further divided into areas of use. Dividing the IP addresses assists in troubleshooting since given the address determines what area we start our troubleshooting process with.

Subnet	IP addressing
Public address	2.2.2.1
router – firewall connection	10.2.1.0/30
firewall – internal firewall	192.168.1.0/24
general internal network	192.168.4.0/24
management network	192.168.3.0/24
VPN network – sales	192.168.10.192/28
VPN network – executive	192.168.12.192/28
VPN network – support	192.168.15.129/28
VPN peer network	192.168.20.129/28

The following table shows the IP addresses for the major systems.

Host	IP address
Router public interface	2.2.2.1
Router private interface	10.2.1.1
firewall outside interface	10.2.1.2
firewall inside interface	192.168.1.1
Public Web server	192.168.1.2
Public Email server	192.168.1.4
secondary firewall outside	192.168.1.7
secondary firewall inside	192.168.4.4
Production app. server	192.168.4.131
Main server	192.168.4.62
IDS	192.168.4.7
Workstations	192.168.4.20 – 60
ISP DNS Server	2.2.0.10
ISP DNS Server	2.2.0.8

Static NAT is used at the border router since we only have one public IP address. The border router maps the public IP address to one NAT'ed private IP address.

Security Policy and tutorial

Border router

The border router is the first component is that traffic sees upon entering the Rune network. Conversely, it is also the last device traffic sees when leaving the Rune network. The purpose of the border router is to filter out the noise and easy traffic. It also provides ingress and egress filtering. This router provides stateless packet filtering.

The border router used is a Cisco 1720 router with a WIC-ENET Ethernet module

connected to the primary firewall. The router is running the latest version of Internetworking Operating System (IOS), version 12.3(5b). This version is required to support the Ethernet module but IOS features IP PLUS and 3DES require 48 Mb RAM while our router only has 32. In the future, Rune plans to upgrade the router memory. Not having IP PLUS and 3DES means that we cannot use network time protocol (ntp) nor `ssh`. This will be investigated for a future design revision.

The router itself is hardened[9] [10], all unnecessary services not needed are disabled.

The routers external interface contains our only public IP address. The router performs Network Address Translations (NAT) for our network. That is all traffic leaving our network all have the public IP address of our router's external interface.

Using NAT affects the types of VPN protocols we can use. We cannot use the Authentication Header (AH) protocol since the protocol uses both the source and destination IP address in calculating a checksum. NAT changes the source address, when going outbound, thus invalidating the checksum. AH, also, only provides data integrity and not data confidentiality. Since, we are interested in confidentiality we will use Encapsulating Security Payload (ESP) protocol. ESP also provides a small amount of data authentication[11].

Router access is allowed only from the inside through the firewall. Access to the router is only enabled for one administrative system. Access is telnet, limited by router memory constraints, at this time but soon will be upgraded to ssh when additional router memory is installed.

Our approach to the router configuration is any configuration not needed or understood is turned off and disabled. Only services required are turned on and enable. In addition, we do not rely on the fact that some of these settings are default settings for the particular version of IOS. The default settings have been known to change depending on which version of IOS we are running.[12]

The following sections describe the routers configuration file and reasons for enabling or disabling each service. The complete configuration file from the border router can be found in appendix III. If a service is not understood it is disabled. The way we can determine if the service is required. Comments, indicated by the ! character, are used in the configuration file to assist the support person in describing the role of the service.

Global configuration commands

```
no service pad
```

This commands disables the packet assembler dis-assembler (PAD) X.29 related commands used on X.25 connections. This service is not used and eliminates another service for the attacker to concentrate on.

```
no service config
```

This command disables the auto loading of the router configuration from a network server. This eliminates a possible avenue for an attacker to install their custom configuration file on our router. This command is used with `boot host` or `boot network` neither which we have enabled.

```
no service tcp-small-servers
```

This command disables the TCP ports that have been used in the past for diagnosis. This includes the `echo, discard, chargen` and `daytime` services. This service is not used and eliminates another service for the attacker to concentrate on.

```
no service udp-small-servers
```

This command disables the UDP ports that have been used in the past for diagnosis. This include the `echo, discard` and `chargen` services. This service is not used and eliminates another service for the attacker to concentrate on.

```
no service finger
```

This command has been replaced by `no ip finger` command.

```
no service dhcp
```

This command disables the Dynamic Host Configuration Protocol (dhcp) and relay agent feature on the router. Since all IP addresses at our perimeter are static this eliminates the router providing automatically a valid IP for the attacker or relaying the service through the router.

```
no ip finger
```

This command disables the finger protocol, rfc 1288. We do not need to provide any information to a remote user. This service is not used and eliminates another service for the attacker to concentrate on

```
ip subnet-zero
```

This command disables the use of subnet 0 for interface addresses and routing updates. This service is not used and eliminates another service for the attacker to concentrate on.

```
no ip classless
```

This command disables the forwarding of packets destined for unrecognized subnets of directly connected networks. With this command disabled the router will discard packets that it does not have a default route for. This service is not used and eliminates another service for the attacker to concentrate on.

```
no ip source-route
```

This command disables the handling of packets with source routing header options. Packets with this option set are discarded. This eliminates the possibility of an attacker specifying a route of their choice.

```
no ip http server
```

This command disables the http server on the router that can be used for configurations. This is not our preferred method for updating our router configurations and thus is disabled to reduce the services running on the router. This service is not used and eliminates another service for the attacker to concentrate on.

```
no ip bootp server
```

This command disables the bootp protocol on the router since we do not use this functionality. This service is not used and eliminates another service for the attacker to concentrate on.

```
no ip domain lookup
```

This command disables the translation or IP addresses to host names. This reduces the risk of an attacker re-directing traffic.

```
no ip cef
```

This command disables Cisco Express Forwarding which controls whether voice is switched on the router. This feature is not used and reduces un-necessary services that allow an attacker to exploit.

```
no cdp run
```

This command disables the Cisco Discovery Protocol (CDP) since it is not required and gives out too much information for the attacker to use to determine our network infrastructure.

```
no snmp-server
```

This command disables all version of Simple Network Management Protocol (SNMP) SNMP can be used to gather information about a device and also control the operation of the device. This ability is weakly protected by a community string on some versions. This service is not used and eliminates another service for the attacker to concentrate on.

```
no boot network
```

This command disables the loading of the router configuration from the network. This eliminates a possible avenue for an attacker to install their custom configuration file on our router.

```
no boot host
```

This command disables the loading of the router configuration from a given host. This eliminates a possible avenue for an attacker to install their custom configuration file on our router.

```
no aaa new-model
```

This command disables the authentication, authorization, and accounting (AAA) access model. This reduces the possibility of an attacker substituting their own authentication model.

```
interface Ethernet0
 ip address 10.2.1.1 255.255.255.252
 ip access-group 120 in
 half-duplex
 no ip unreachable
 no ip mask-reply
```

```
 no ip redirects
 no ip directed-broadcast
 no ip proxy-arp
 no cdp enable
 ip nat inside
!
interface FastEthernet0
 ip address 2.2.2.1 255.255.0.0
 ip access-group 110 in
 no ip unreachable
 no ip mask-reply
 no ip redirects
 no ip directed-broadcast
 no ip proxy-arp
 no cdp enable
 ip nat outside
 speed auto
```

The following commands are applied against each interface, FastEthernet0 and ethernet0 as shown above. The interface commands also allow for the IP address configuration and the applying of the Access Control Lists (ACL).

```
ip nat [inside | outside ]
```

This command enables NAT on our inside and outside interfaces. This allows the router to have one public address IP address on the external interface and private address on the internal interface. This command is used in conjunction with the `ip nat in|outside source` command.

```
no ip unreachables
```

This command disables the sending out of ICMP unreachable packets. This reduces the amount of information leaked out of our network infrastructure. This reduces the information available for the attacker to gather.

```
no ip mask-reply
```

This command disables sending out a response to an ICMP request for our subnet mask. This reduces the information we give away about of network infrastructure.

```
no ip redirects
```

This command disables the sending of ICMP redirect packets for packets that are sent out the same interface it was received. This eliminates a possible avenue for the attacker to reroute traffic to sites of their choice.

```
no ip directed-broadcast
```

This command disables the translation of a directed broadcast to physical addresses. This command reduces the risk of being used as an amplifier for attackers.

```
no ip proxy-arp
```

This command disables the proxy ARP on the interface. Not sure of how this command is used therefore it is disabled.

```
no cdp enable
```

This command disables the CDP on this interface. This information can only assist attackers in gathering information about our network.

```
no service tcp-keepalives-in
no service tcp-keepalives-out
```

The command disables the generation of keep alive packets for both in and out bound connections.

```
service timestamps debug uptime
service timestamps log uptime
```

This command puts router uptime into log messages. This assists support persons if the router has unexpectedly been restarted.

```
service password-encryption
```

Ensure passwords are encrypted when viewing the passwords in the configuration file.

```
hostname vax1
```

Select a non descriptive host name for the router.

```
banner login # Unauthorized access is prohibited #
```

Add a banner warning unauthorized users that access is prohibited.

```
ip domain-name rune.ca
```

This command appends the domain name to unqualified hosts. Since DNS is not enabled on the router this is used for informational documentation only.

```
boot-start-marker
boot system flash
boot-end-marker
```

Configures the router to use flash memory for booting the system image. This reduces the risk of an attacker booting their own custom image on our router. `boot-start-marker` and `boot-end-marker` are added by the router itself.

```
enable secret 5 $1$strongS%@passwdW71
```

The command specifies an additional layer of security over the `enable` password command.

```
memory-size iomem 25
```

The command specifies the amount of memory set aside for I/O. This is the default value.

```
clock timezone EST -5
```

The command set the time zone for display purposes since internally the time is kept in Universal Coordinated Time (UTC).

```
logging buffered 4096 debugging
logging history notifications
logging trap debugging
logging origin-id string c_rtr
logging facility local4
logging source-interface Ethernet0
logging 10.2.1.2
```

The logging section is where we configure our router to log to our syslog server. Our syslog server needs to be configured to accept these remote syslog `udp` packets. We specify a unique string to identify the router with to assist in log analysis. We started off with debug level logging and then reduce the level to warnings.

The following section describes our ACLs for the border router. They have the following format.[13]

```
access-list id {permit | deny} protocol source sport dest dport options
```

`access-list`
Configuration mode command to create an ACL

`id`
A number that indicated the type of ACL used. 1 to 99 are standard ACL's that can only test for IP source address. Extended ACL's are in the 100 to 199 range and can test for source, destination address, protocol, ports and icmp types.

`permit | deny`
Either allow (`permit`) or not (`deny`) the packet to traverse the router

`protocol`
The name or number of the IP protocol. `IP` covers any of the following: `icmp`, `ip`, `tcp` or `udp`.

`source, dest`
The source and destination address including subnet mask. The subnet mask used is the traditional Cisco wildcard notation.

Cisco ACLs are processed in a top down order. When a match is made the processing finishes and no other ACLs are evaluated. If no match is made our last ACL denies the packet. Our ACL approach is to handle the specifics and then handle more general situations in later ACLs. Also included are egress and ingress filtering.

The following set of ACLs are applied to our external, public interface.

```
access-list 110 deny ip 192.168.0.0 0.0.255.255 any log
access-list 110 deny ip 172.16.0.0 0.15.255.255 any log
access-list 110 deny ip 10.0.0.0 0.0.0.255 any log
```

Block and log any rfc 1918 private addresses.

```
access-list 110 deny ip 224.0.0.0 15.255.255.255 any log
```

Block any multicast traffic.

```
access-list 110 deny ip 127.0.0.0 0.255.255.255 any log
```

Block any loopback traffic.

```
! deny iana   dated: 2004-01-15
! source www.iana.org/assignments/ipv4-address-space
access-list 110 deny ip 0.0.0.0 0.255.255.255 any log
access-list 110 deny ip 255.255.255.255 0.0.0.0 any log
```

Block broadcast traffic. The zero subnet is also part of the IANA reserved addresses[14]. Rune has decided not to specifically block these addresses since this would mean adding about 30 more ACLs and making is harder to maintain. Also, we would not have to periodically review the IANA web site to determine if a new address block has been allocated.

```
! deny incoming traffic that states being form our firewall
! since this is our externel interface and the firewall is
! on our internel interface
access-list 110 deny ip 10.2.1.2 0.0.0.255 any log
```

Block traffic that states it is from our firewall interface.

```
!
! BLACK LISTED hosts and networks
!
! gotomypc.com - 66.151.158.183
access-list 110 deny ip 66.151.158.183 0.0.0.255 any log
```

Block and log access to this firewall bypassing site. This section will be used to add further hosts and networks that we do not want to deal with. Future option, could look at a real time black hole service when we install our own DNS servers.

```
!  DENY ACL HERE
! block stuff that could leak out!
! windows then unix stuff then others

! windows stuff, netbios, ds, mssql
```

```
access-list 110 deny tcp any any range 135 139 log
access-list 110 deny udp any any range 135 139 log
access-list 110 deny tcp any any eq 445 log
access-list 110 deny udp any any eq 445 log
access-list 110 deny tcp any any eq 1443 log
access-list 110 deny udp any any eq 1443 log
```

Block any of the Windows protocols from entering our network. Depending on the volume of the logging we may not log all of the above protocols all the time.

```
!unix stuff
!finger
access-list 110 deny tcp any any eq 79 log
access-list 110 deny udp any any eq 79 log
!tftp
access-list 110 deny udp any any eq 69 log
!rpc
access-list 110 deny tcp any any eq 111 log
access-list 110 deny udp any any eq 111 log

!snmp
access-list 110 deny tcp any any range 161 162 log
access-list 110 deny udp any any range 161 162 log
!r* utils
access-list 110 deny tcp any any range 512 514 log
!nfs
access-list 110 deny tcp any any eq 2049 log
access-list 110 deny udp any any eq 2049 log
! x windows
access-list 110 deny tcp any any range 6000 6255 log
```

Block any of the popular Unix protocols from entering our network. We should especially not allow any syslog traffic from the outside into our syslog servers.

```
access-list 110 permit icmp any any 3 4
access-list 110 deny icmp any any log
```

Block all un-necessary `icmp` traffic. Only allow fragmentation required to be able to handle mtu changes.

```
! ALLOW STUFF follows
!
access-list 110 permit tcp any 10.2.1.2 0.0.0.0 eq 80
access-list 110 permit tcp any 10.2.1.2 0.0.0.0 eq 443
```

```
access-list 110 permit tcp any 10.2.1.2 0.0.0.0 eq 25
```

Allow any host, other than specifically blocked above, access to our public web, email servers.

```
! allow stuff out
!access-list 110 permit tcp 10.2.1.2 0.0.0.0 any
established
access-list 110 permit tcp any any established log
```

Allow any already established connection back into our network. That is a connection with the ACK, FIN, PSH, RST, SYN or URG control bits set.

```
! let dns go by since fw will check state
! only dns from our isp's dns servers
access-list 110 permit udp 2.2.0.8 0.0.0.0 eq 53 2.2.2.1 0.0.0.0
access-list 110 permit udp 2.2.0.10 0.0.0.0 eq 53 2.2.2.1 0.0.0.0
```

Allow DNS traffic, from our ISPs DNS server, past the border router. The stateful firewall will determine if the traffic is a response based on a request from our network.

```
! vpn check specifics
access-list 110 permit udp any 2.2.2.1 0.0.0.0 eq 500 log
access-list 110 permit esp any 2.2.2.1 0.0.0.0 log
```

Allow VPN traffic into our network. It is logged to ensure authorized persons are connecting to our VPN.

```
access-list 110 deny ip any any log
```

Block and log any traffic that is not handled.

The following set, 120, of ACLs performs our egress filtering on the internal interface.

```
access-list 120 deny ip 192.168.0.0 0.0.255.255 any log
access-list 120 deny ip 172.16.0.0 0.15.255.255 any log
access-list 120 deny ip 10.0.0.0 0.0.0.255 any log
```

Block and log any rfc 1918 private addresses from leaving our network.

```
access-list 120 deny ip 224.0.0.0 15.255.255.255 any log
```

Block and log any multicast traffic from leaving our network.

```
access-list 120 deny ip 127.0.0.0 0.255.255.255 any log
```

Block and log any loopback traffic from leaving our network.

```
access-list 120 deny ip 0.0.0.0 0.255.255.255 any log
```

Block and log any broadcast traffic from leaving our network. As with the exterior interface we do not specifically block IANA reserved addresses.

```
! deny traffic that states it is from the outside
access-list 120 deny ip 2.2.2.1  0.0.0.0 any log
```

Block and log any traffic that states it is from our public IP address.

```
access-list 120 deny icmp any any log
```

Block and log all icmp traffic from leaving our network. Do not need to allow fragmentation required since we control the networks on the inside.

```
!  DENY ACL HERE
! block stuff that could leak out!
! windows then unix stuff then others

! windows stuff netbios, ds, mssql
access-list 120 deny tcp any any range 135 139 log
access-list 120 deny udp any any range 135 139 log
access-list 120 deny tcp any any eq 445 log
access-list 120 deny udp any any eq 445 log
access-list 120 deny tcp any any eq 1443 log
access-list 120 deny udp any any eq 1443 log
```

Block and log any of the Windows traffic from leaving our network. All of this traffic is also blocked by the default rule on the firewall.

```
!unix stuff
!finger
access-list 120 deny tcp any any eq 79 log
access-list 120 deny udp any any eq 79 log
```

```
!tftp
access-list 120 deny udp any any eq 69 log
!rpc
access-list 120 deny tcp any any eq 111 log
access-list 120 deny udp any any eq 111 log
!snmp
access-list 120 deny tcp any any range 161 162 log
access-list 120 deny udp any any range 161 162 log
!r* utils
access-list 120 deny tcp any any range 512 514 log
!nfs
access-list 120 deny tcp any any eq 2049 log
access-list 120 deny udp any any eq 2049 log
! x
access-list 120 deny tcp any any range 6000 6255 log
```

Block and log any of the popular Unix traffic from leaving our network. All of this traffic is also blocked by the default rule on the firewall.

```
! ALLOW STUFF follows
!
access-list 120 permit tcp 10.2.1.2 0.0.0.0 any eq 80
access-list 120 permit tcp 10.2.1.2 0.0.0.0 any eq 443
access-list 120 permit tcp 10.2.1.2 0.0.0.0 any eq 25
access-list 120 permit tcp 10.2.1.2 0.0.0.0 any eq 22

! only dns to our isp's dns servers
access-list 120 permit udp 10.2.1.2 0.0.0.0 2.2.0.10 0.0.0.0 eq 53
access-list 120 permit udp 10.2.1.2 0.0.0.0 2.2.0.8 0.0.0.0 eq 53


! vpn check specifics
access-list 120 permit udp 10.2.1.2 0.0.0.0 any eq 500
access-list 120 permit esp 10.2.1.2 0.0.0.0 any
access-list 120 permit udp 10.2.1.2 0.0.0.0 142.217.253.235 0.0.0.0 eq
500
access-list 120 permit esp 10.2.1.2 0.0.0.0 142.217.253.235 0.0.0.0
```

Allow VPN related traffic out. The first ACL allows to any address. This is related to remote access users. The second ACL is not really needed but it is used to document the VPN peer connection.

```
access-list 120 permit tcp any any established log
```

Allow any previous established tcp connections back in.

```
access-list 120 deny ip any any log
```

Block and log any traffic that does not meet the above rules.

```
access-list 100 permit tcp host 192.168.1.7 eq 23 log
access-list 100 deny ip any any log
```

Restrict telnet access to the router itself only from the management systems as well as log both successfully and unsuccessfully connections.

```
line con 0
 session-timeout 5
 exec-timeout 5 0
 password 7 dXPaSSwDx
 login
```

Add session timeouts on the console port. This automatically disconnects the session, if not activity is received after 5 minutes, that is, if the support person is distracted. A console password is also added.

```
! disable aux port no exec
line aux 0
 login local
 no exec
 no password
```

Disable the auxiliary port.

```
line vty 0 4
 access-class 100 in
 session-timeout 5
 exec-timeout 5 0

 password 7 dXPaSSwDx
 login
```

Add session timeouts on the telnet port. This automatically disconnects the session if the support person is distracted and reduces exposure of some one changing the configuration without approval. Also, apply the telnet ACL.

Primary Firewall

The primary firewall is the next component traffic encounters when entering the Rune network. The primary firewall is a Cisco PIX 501 dedicated firewall appliance running the latest version of the PIX Firewall software, version 6.3(1).

The PIX 501 serves as both the firewall and VPN gateway for Rune Enterprises. The decision to use the PIX for both functions was based on finances and the requirement to keep the infrastructure straightforward. Since Rune Enterprise only has two traveling sales people, IT support staff and a VPN tunnel to the supplier it was not deemed worth the cost and complexity to design a dedicated VPN device. The 501 can handle up to 10 VPN connections.

Since the PIX 501 does not support more than two interfaces the mail and web servers are accessed through port forwarding. The clear text throughput of 60Mbs with 7500 concurrent connections is ample to meet our current needs and for some time in to the future.

Our approach to the firewall configuration is similar to the router configuration. That is, any configuration not needed or understood is turned off and disabled. Only services required are turned on and enabled. In addition, we do not rely on the fact that some of these setting are default setting for the particular version of PIX firewall software. Most of these settings are the default ones along with the addition of ACLs.

Firewall access is allowed only from the inside. Access to the firewall is only enabled for one administrative system. Access is ssh version 1 which offers better protection than telnet.

The firewall itself is hardened and kept, along with the router and other network gear, in a locked room with limited access.

To allow users to determine if “internet access” is up and available we allow ping access to the firewall only. We do not allow ping access outside our network.

The following are the firewall ACLs. More details regarding the initial configuration of the PIX can be found in the tutorial section.

The format of the firewall ACL for the PIX is as follows[15]:

```
access-list id {permit | deny} protocol source-addr destination-addr
```

`access-list`
Configuration mode command to create an ACL

`id`
Name of the ACL, either a number or a name

`permit | deny`
Either allow (`permit`) or not (`deny`) the packet to traverse the firewall

`protocol`
The name or number of the IP protocol. `IP` covers any of the following: `icmp`, `ip`, `tcp` or `udp`.

`source-addr, destination-addr`
The source and destination address including subnet mask. The subnet mask used in the PIX is the traditional notation versus the inverse wildcards used in routers.

```
interface ethernet0 auto
interface ethernet1 100full
```

Setting up the parameters for the firewall interfaces.

```
nameif ethernet0 outside security0
nameif ethernet1 inside security100
```

Naming the interfaces and setting the security levels. These security levels determine what required commands need to be present.

```
enable password 8Rypass7WORD24 encrypted
```

Ensure the enable password is encrypted and well protected.

```
passwd 2KfstrongNIpasswdI.2KYOU encrypted
```

Ensure the telnet password is encrypted and well protected.

```
hostname vax8
domain-name rune.ca
```

Name the firewall something non descriptive and set the IPSEC domain name for Certificate authorities used for `ssh` rsa key pairs.

```
clock timezone EST -5
clock summer-time EDT recurring
```

Time keeping setup.

```
fixup protocol http 80
fixup protocol smtp 25
fixup protocol esp-ike
```

Use of Cisco's Adaptive Security Algorithm (ASA) for the supported protocols. We disable the protocols, listed below, not used that are enabled by default.

```
no fixup protocol ftp 21
no fixup protocol h323 h225 1720
no fixup protocol h323 ras 1718-1719
no fixup protocol ils 389
no fixup protocol rsh 514
no fixup protocol rtsp 554
no fixup protocol sip 5060
no fixup protocol sip udp 5060
no fixup protocol skinny 2000
no fixup protocol sqlnet 1521
```

```
filter activex 80 0 0 0 0
filter java 80 0 0 0 0
```

Block both activeX and java.

```
no names
```

Do not allow use of names to be associated with an IP address.

```
: for VPN to bypass nat id 0
access-list in_out_nat0_acl permit ip any 192.168.5.192   255.255.255.224 log
access-list in_out_nat0_acl permit ip any 192.168.10.192  255.255.255.224 log
access-list in_out_nat0_acl permit ip any 192.168.12.192  255.255.255.224 log
access-list in_out_nat0_acl permit ip any 192.168.15.192  255.255.255.224 log

: dynamic acls
access-list out_crypto_dyn_20 permit ip any 192.168.5.192 255.255.255.224 log
access-list out_crypto_dyn_20 permit ip any 192.168.10.192 255.255.255.224 log
access-list out_crypto_dyn_20 permit ip any 192.168.12.192 255.255.255.224 log
access-list out_crypto_dyn_20 permit ip any 192.168.15.192 255.255.255.224 log

: VPN peer!
access-list out_20 permit ip any 5.23.16.224 255.255.255.240 log
```

The above ACLs are used for VPN remote access. The first set indicates to bypass the NATing of the incoming remote access address. The second set is the dynamic ACL that allows our remote access user access from any IP address to the VPN pool addresses.

The following ACLs are applied on the outside interface. The ACLs follow basically the same approach for the border router with regards the non routeable addresses.

```
:
: FIREWALL EXTERNAL INTERFACE
:
: log since rtr should stop these!

:  access-list out-in deny ip 10.0.0.0 255.0.0.0 any log
access-list out-in deny ip 172.16.0.0 255.240.0.0 any log
access-list out-in deny ip 192.168.0.0 255.255.0.0 any log
access-list out-in deny ip 224.0.0.0 240.0.0.0 any log
access-list out-in deny ip 127.0.0.0 255.0.0.0 any log
access-list out-in deny ip 0.0.0.0 255.0.0.0 any log
access-list out-in deny ip 255.255.255.255 255.255.255.255 any log
```

Deny access to our network from the private addresses. These should have been already blocked by our border router.

```
access-list out-in permit tcp any host 10.2.1.2 eq 80
access-list out-in permit tcp any host 10.2.1.2 eq 443
access-list out-in permit tcp any host 10.2.1.2 eq 25
```

Allow access to our public services, web and email.

```
access-list out-in permit udp host 10.2.1.1 host 10.2.1.2 eq syslog
```

Allow syslog packets from our router into our network.

```
access-list out-in permit udp host 2.2.0.10 host 10.2.1.2 eq 53
access-list out-in permit udp host 2.2.0.8 host 10.2.1.2 eq 53
```

Allow DNS replies, only from our ISPs name servers, back into our network.

```
access-list out-in deny ip any any log
```

Our default deny and log anything that is not matched by a previous ACL.

The following ACLs apply to the firewalls inside interface

```
:
: FIREWALL internal INTERFACE
:
: egress
access-list in-out deny ip 10.0.0.0 255.0.0.0 any log
access-list in-out deny ip 172.16.0.0 255.240.0.0 any log
```

```
:  access-list in-out deny ip 192.168.0.0 255.255.0.0 any log

access-list in-out deny ip 224.0.0.0 240.0.0.0 any log
access-list in-out deny ip 127.0.0.0 255.0.0.0 any log
access-list in-out deny ip 0.0.0.0 255.0.0.0 any log
access-list in-out deny ip 255.255.255.255 255.255.255.255 any log
```

Deny exit to any private addresses with the exception of our NAT'd network.

```
: allow to outside web sites, http, https
access-list in-out permit tcp 192.168.1.0 255.255.255.0 any eq 80
access-list in-out permit tcp 192.168.1.0 255.255.255.0 any eq 443
```

Allow our users access to outside web sites.

```
: only allow mail access to our mail relay
access-list in-out permit tcp 192.168.1.0 255.255.255.0 host 192.168.1.4 eq 25
```

Only allow mail access through our mail server.

```
: ADMIN stuff ( ssh out )
access-list in-out permit tcp 192.168.1.0 255.255.255.0 any eq 22
```

Allow `ssh` out for the support team.

```
access-list out-in permit udp any host 2.2.0.10 eq dns
access-list out-in permit udp any host 2.2.0.8 eq dns
```

Allow access to our ISPs DNS servers.

```
access-list in-out deny ip any any log
```

Our default deny rule.

```
banner exec Unauthorized use prohibited
```

Set up a warning banner.

```
logging on
logging buffered warnings
logging trap warnings
logging device-id string p_fw
logging host inside 192.168.1.7
```

Ensure we have sufficient logging to our syslog server on the inside. Add a unique identifier to assist in log analysis. Also, log to the internal buffer.

```
mtu outside 1500
mtu inside 1500
```

Setting of the Maximum Transmittion Unit (MTU) for Ethernet.

```
ip address outside 10.2.1.2 255.255.255.252
ip address inside 192.168.1.1 255.255.255.0
```

Setting the IP address for both interfaces

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

Sets up default actions for Cisco's IDS. The alarm action reports to our syslog server that a signature match was detected in a packet.

```
ip local pool vpnexec  192.168.10.200-192.168.10.220
ip local pool vpnsales 192.168.12.200-192.168.12.220
ip local pool vpnsup   192.168.15.200-192.168.15.220
```

Set up the DHCP pool for VPN remote access groups. Each group has its own IP address range and its own different unique group password.

```
arp timeout 14400
```

Address resolution protocol (arp) default timeout value.

```
global (outside) 1 interface
```

Setting up our NAT policy, one to many, to be Port Address translation (PAT). This states all internal IP addresses are mapped to our external interface, 10.2.1.2.

```
nat (inside) 0 access-list in_out_nat0_acl
```

Configure the NAT exemption access-list. This access-list is used with VPN access.

```
nat (inside) 1 0.0.0.0 0.0.0.0 0 0
```

NAT all inside addresses traveling outside.

```
static (inside,outside) udp interface 514 192.168.1.2 514  netmask 255.255.255.255 0 0
static (inside,outside) tcp interface www 192.168.1.2 www  netmask 255.255.255.255 0 0
static (inside,outside) tcp interface 443 192.168.1.2 443  netmask 255.255.255.255 0 0
static (inside,outside) tcp interface 25 192.168.1.4 25    netmask 255.255.255.255 0 0
```

This setting, port forwarding, provides us with a simulated DMZ on the firewall that has only two interfaces. Certain services are forwarded to selected host. The disadvantage is that we cannot have more that one host serving a given service.

```
access-group out-in in interface outside
access-group in-out in interface inside
```

Apply the named ACLs to their respective interfaces.

```
route outside 0.0.0.0 0.0.0.0 10.2.1.1 1
```

Setting up our default route to the internal interface of our border router.

```
:nat'd behind 192.168.1.7!!!!
route inside 192.168.4.0 255.255.255.0 192.168.1.7 1
```

Set up a router to our internal firewall for the internal network for the VPN users.

```
timeout xlate 0:05:00
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
```

Default time out values, not changed.

```
clear http
```

Disable the http server on the PIX and thus do not allow the use of Cisco PIX Device Manager (PDM), the web tool to monitor and configure the firewall.

```
no snmp-server location
no snmp-server contact
snmp-server community !@#$%smnp-not-used-^&*()$((asDAF9((^@
no snmp-server enable traps
```

Since we are not using SNMP we disable all that we can. The community string does not appear allowed to be disable therefore we set it to a difficult password using the maximum length allowed at 32 characters.

```
floodguard enable
```

Enable protection against flood attacks.

```
ssh 192.168.1.4 255.255.255.0 inside
ssh timeout 5
```

Enable `ssh` since even version 1 is better than telnet. This is used to access the firewall directly. Detail of the ssh setup are provided in the tutorial.

```
console timeout 5
```

```
clear dhcpd
```

Do not enable the PIX dhcp server since all our addresses are static.

```
terminal width 80
```

Default screen width setting.

```
Cryptochecksum:eed0cd9711b358c8f0d324f0e7061b31
: end
```

This checksum is very important to document since it changes anytime the firewall configuration is updated.

VPN policy

The VPN policy is divided into two sections, the remote access and the peer network.

The following VPN peer policy is used for our partner network. Since our peer network is always connected we chose to use, AES-256, the strongest encryption provided by the PIX. Due to low expected volume we are not concerned about the overhead this requires.

This access is from a known public IP address and a given internal IP range.

They will have restricted network access to our production application server only.

The following VPN policy was used for remote access. Triple DES was selected are being strong enough but not taxing the firewall with processing overhead. Remote access is usually of limited duration and has a user who would be impacted by additional encryption overhead.

Remote access users do not have known IP addresses since they may be connecting from many and varied locations. Each remote access group will have different internal network access.

Since our border router uses static NAT if affects the types of VPN protocols we can use. We cannot use the Authentication Header (AH) protocol since the protocol uses both the source and destination IP address in calculating a checksum. NAT changes the source address, when going outbound, thus invalidating the checksum. AH, also, only provides data integrity and not data confidentiality. Since, we are interested in confidentiality we will use Encapsulating Security Payload (ESP) protocol. ESP also provides a small amount of data authentication[16].

Our two traveling sales people along with our executive management use the Cisco VPN client version 4.0.2(A). The support staff use the Linux version 4.0.3(B). When connected via the VPN neither group has access to the internet either through their laptop or through the Rune network, thus no split-horizon.

The pre-shared secret, the key, has to be well chosen since this is the weakest link.[17] Each group has its own key to minimize the threat.

Since this Rune Enterprises is small company we do not use the certificate for VPN connections due to their complexity at this time.

```
sysopt connection permit-ipsec
```

Allow IPsec traffic to pass through without access-list checking.

```
crypto ipsec transform-set t-aes esp-aes-256 esp-sha-hmac
```

This command sets up the transform sets

```
crypto dynamic-map outside_dyn_map 20 match address out_crypto_dyn_20
crypto dynamic-map outside_dyn_map 20 set transform-set t-aes
```

This command sets up the dynamic crypto map, that is a policy template for remote users with unknown IP addresses. The first line associates an ACL with

the dynamic crypto map. The second line states what transform set we are using, for example AES encryption.

```
crypto map outside_map 20 ipsec-isakmp
crypto map outside_map 20 match address out_20
crypto map outside_map 20 set peer 4.2.2.4
crypto map outside_map 20 set transform-set t-aes
```

This command sets up the static VPN peer parameters. It is similar to the dynamic set up above. We state we are using IKE to set up the security associations. Next we associate an ACL with the IP addresses and set up the peer IP address. Then we state what transforms we are using.

```
crypto map outside_map 65535 ipsec-isakmp dynamic outside_dyn_map
crypto map outside_map interface outside
```

This command sets up the dynamic crypto map, used when requests are not matched by the 20 entry, that is the remote users.

```
isakmp enable outside
```

This commands enables Internet Key Exchange (IKE) on the outside interface.

```
isakmp key #passwd# address 4.2.2.4 netmask 255.255.255.255 no-xauth no-config-mode
```

This command sets up the peer VPN connection. The address listed is the peer's public IP address. Other ACLs reference the peer's internal IP address structure where we have access. `no-xauth` and `no-config-mode` state that the PIX will not ask for a username and password nor try to assign a dynamic IP address.

```
isakmp policy 10 authentication pre-share
isakmp policy 10 encryption 3des
isakmp policy 10 hash sha
isakmp policy 10 group 2
isakmp policy 10 lifetime 86400

isakmp policy 20 authentication pre-share
isakmp policy 20 encryption aes-256
isakmp policy 20 hash sha
isakmp policy 20 group 2
isakmp policy 20 lifetime 86400
```

These commands set up the IKE policy we are using. The number, 10 or 20, sets the priority of the policy. Pre-shared secret is the authentication method for both policies. The first encryption method is triple DES and the second is AES. SHA is used for the hash, for both, since MD5 has shown some cracks and is not used for anything new.[18] Group 2 1024 bit Diffie-Hellman is used for key exchange. Cisco recommends group 5 for AES encryption but Rune tech support has been unable to get the VPN working with group 5, thus group 2 is being used while this problem is resolved.

```
vpngroup vpnsales address-pool vpoolsales2
vpngroup vpnsales idle-time 900
vpngroup vpnsales password mysalestrongpasswd

vpngroup vpnsales address-pool vpoolsales1
vpngroup vpnsales idle-time 900
vpngroup vpnsales password mysalestrongotherpasswd

vpngroup vpnexec address-pool vpoolexec
vpngroup vpnexec idle-time 900
vpngroup vpnexec password myexecstrongpasswd

vpngroup vpnsup address-pool vpoolsup
vpngroup vpnsup idle-time 900
vpngroup vpnsup password mysupstrongpasswd
```

This sets up the VPN group each having their own password as well as their own IP address range. This division assists tech support when troubleshooting is required. Idle time out has been set to 15 minutes or 900 seconds.

Pix 501 tutorial

This tutorial section describes how to set up and configure a Cisco PIX 501 firewall appliance acting as a firewall. VPN functionality will not be covered in depth.

The first step is to ensure you can communicate with the serial console port. Use an appropriate terminal emulation program, for example Kermit on a linux 9.0 system. The serial port on you computer should be set to the following: 9600 8N1, hardware. Communication speed of 9600 baud, 8bit bytes, no spot bit and one parity bit using hardware flow control.

Once this is done connect the supplied cable to the firewall and your computer serial port and then plug in the power cord for the PIX. You should see the following messages with a few seconds.

```
CISCO SYSTEMS PIX-501
Embedded BIOS Version 4.3.200 07/31/01 15:58:22.08
Compiled by morlee
16 MB RAM

PCI Device Table.
Bus Dev Func VendID DevID Class              Irq
 00  00  00   1022   3000  Host Bridge
 00  11  00   8086   1209  Ethernet           9
 00  12  00   8086   1209  Ethernet           10

Cisco Secure PIX Firewall BIOS (4.2) #6: Mon Aug 27 15:09:54 PDT 2001
Platform PIX-501
Flash=E28F640J3 @ 0x3000000

Use BREAK or ESC to interrupt flash boot.
Use SPACE to begin flash boot immediately.
Reading 1921536 bytes of image from flash.
########################################################################
#########
#########################
16MB RAM
mcwa i82559 Ethernet at irq  9  MAC: 000d.eeee.eee1
mcwa i82559 Ethernet at irq 10  MAC: 000d.eeee.eee2
Flash=E28F640J3 @ 0x3000000
BIOS Flash=E28F640J3 @ 0xD8000

  ------------------------------------------------------------------------
-
                                ||        ||
                                ||        ||
                               ||||      ||||
                           ..:||||||:..:||||||:..
                          c i s c o S y s t e m s
                         Private Internet eXchange
  ------------------------------------------------------------------------
-
                         Cisco PIX Firewall

Cisco PIX Firewall Version 6.3(1)
Licensed Features:
Failover:           Disabled
VPN-DES:            Enabled
VPN-3DES-AES:       Enabled
Maximum Interfaces: 2
Cut-through Proxy:  Enabled
Guards:             Enabled
URL-filtering:      Enabled
Inside Hosts:       50
Throughput:         Unlimited
IKE peers:          10

  ****************************** Warning ******************************
  Government Warning
  ****************************** Warning ******************************
  Legalese stuff..
```

```
.outside interface address added to PAT pool
.
Cryptochecksum(unchanged): c15f9fd5 30c2956f 7bca325f 0625ba7f
Type help or '?' for a list of available commands.
pixfirewall>
```

At this point the PIX is booting with a factory default configuration file. This configuration can be found in appendix I. This default configuration meets the needs of most SOHO users.

The Cisco PIX Device Manager (PDM), version 3.0(1) is a great tool to get you started and to provide ongoing monitoring of the firewall. It is great to assist you in setting up VPNs with its VPN wizard.

Also, a default, are the interface setup and security levels. This allows traffic to flow from the high security level to the lower security level but not the reverse. These security levels determine what required commands need to be present. Access from a higher security level to a lower requires the `nat` and `global` commands, while lower to higher require `static` and `access-list` commands.

```
interface ethernet0 auto
interface ethernet1 100full
nameif ethernet0 outside security0
nameif ethernet1 inside security100
```

One of the first things you need to do is set the enable password. The minimum length is 3 but your should be at least 8 with upper and lower case characters as well as some special characters. The enable mode is a privileged mode that allows you to change the configuration of the firewall. If you already have an encrypted password you can add the `encrypted` keyword.

```
enable password my#$%strong&*$PaSSwd!
```

Sets the enable password.

```
passwd my@#telNET@#passWD))
```

Sets the telnet password which if not set is “cisco.”

At this point you can I set up a tftp server so I could write out the configuration file to a local UNIX server and use my favorite editor to build the configuration file. This will ensure the configuration file has your passwords and not the defaults.

```
vax8# write net 192.168.1.2:pix.conf
Building configuration...
TFTP write 'pix.conf' at 192.168.1.2 on interface 1
```

```
[OK]
vax8#
```

When you have made your changes to the configuration file and are ready to test them I « cleared » the running configuration and then reloaded my updated configuration. This way I knew that I was not appending ACLs to existing ACLs. The steps I took are listed below.

```
vax8(config)# clear config all
pixfirewall(config)# ip address inside 192.168.1.1
pixfirewall(config)# config net 192.168.1.2:pix.conf
..outside interface address added to PAT pool
..
Cryptochecksum(changed): c34e12fe acce8ffe dca42430
7302b9ea
Config OK
```

A couple of notes regarding the above sequence: The clearing of the configuration has to been done in `config` mode as well as the other commands. When the configuration is cleared the firewall reverts back to defaults as seen by the change in hostname. Also, we need to give the firewall its internal IP address back since it does not have one once we clear the configuration. We are then ready to load the new configuration file. This method of clearing and reloading the configuration file allows you an easy method to develop and test your firewall rule base.

The following commands are used to setup the PIX to use `ssh` version 1. This step should be done as one of your last steps prior to production testing. The reason is the `clear config all` also erases the private key on the PIX.

```
ca generate rsa key 1024
show ca mypubkey rsa
ca save all
```

The first step is to generate a key. The second step just verifies that the key has been successfully generated. The last step saves the information. The key is based on the hostname and the default domain so these need to be set prior to executing the commands.

As shown, above, we set the inside firewall address to be 192.168.1.1. You may change this depending on your requirements. For the purpose of the tutorial the inside network will be 192.168.1.0/24 with the firewall inside address 192.168.1.1. The outside IP address will be set to 10.2.1.2. All changes shown below can be made using your favorite editor and then uploading the new configuration as shown above.

```
hostname vax8
```

```
domain-name rune.ca
```

Give your PIX a non descriptive name and also set the domain name. The host name with the domain are used for the RSA keys for `ssh` access.

```
fixup protocol ftp 21
fixup protocol http 80
fixup protocol smtp 25

no fixup protocol h323 h225 1720
no fixup protocol h323 ras 1718-1719
no fixup protocol ils 389
no fixup protocol rsh 514
no fixup protocol rtsp 554
no fixup protocol sip 5060
no fixup protocol sip udp 5060
no fixup protocol skinny 2000
no fixup protocol sqlnet 1521
```

By default all of the above protocols are enabled. In our network we only need the `http` and `smtp` protocols so we disabled the remaining protocols that are not required by stating, for example, `no fixup protocol ils 389`. The fixup protocol command enables Cisco's Adaptive Security Algorithm (ASA) for the given protocols. This is basically the stateful inspection of packets for the given protocol.

The format of the firewall ACL for the PIX is as follows[19]:

```
access-list id {permit | deny} protocol source-addr destination-addr
log
```

`access-list`
Configuration mode command to create an ACL

`id`
Name of the ACL, either a number or a name

`permit | deny`
Either allow (`permit`) or not (`deny`) the packet to traverse the firewall

`protocol`
The name or number of the IP protocol. `IP` covers any of the following: `icmp`, `ip`, `tcp` or `udp`.

`source-addr, destination-addr`

The source and destination address including subnet mask. The subnet mask used in the PIX are the traditional notation versus the inverse wildcards used in routers.

`log`
Indicates that you want the match to be logged.

Creating ACLs are based the security policy you are trying to implement. See appendix II**,** for an example of the ACLs used for Rune Enterprises.

The PIX allows you to use either the port number or the well known name to indicate services on port. If you use a name ensure that the service you think you are checking is really the one you mean. For example `dnsix` is not port 53 but port 90. `domain` is port 53.

Another point to watch out for is where you put the port number when you are creating an ACL. For example, the first ACL below, you think you are allowing access to the ISPs DNS, 2.2.0.10, from our outside interface, 10.2.1.2, but the second ACL is the correct way to write that rule.

wrong
```
access-list out-in permit udp host 2.2.0.10 host 10.2.1.2 eq 53
```

correct
```
access-list out-in permit udp host 2.2.0.10 eq 53 host 10.2.1.2
```

```
access-list out-in deny ip any any log
```

Ensure that you have a default rule that denies any packet that does not match a previous one. Writing the rules for the internal interface is similar to the process above.

```
logging on
logging trap warnings
logging device-id string p_fw
logging host inside 192.168.1.2
```

To enable logging to a central syslog server, the first step is to enable logging. The next step is to select a logging level. I started out with the debugging level to assist in the development of the rule based and then I turned down the logging level to warning. Setting a `device-id` for the logging messages assists with log analysis. If you want to log `ftp` commands and web site URLs you need to set the logging level to debugging.

```
ip address outside 10.2.1.2 255.255.255.252
```

```
ip address inside 192.168.1.1 255.255.255.0
```

Set up out interface addresses.

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

Sets up default actions for Cisco's IDS. These settings are not changed. The alarm action reports to our syslog server that a signature match was detected in a packet.

```
:clear global to fac defa reset prior to reconfig
clear global
global (outside) 1 interface

: clear nat for fac default no erro msg
clear nat
nat (inside) 0 access-list in_out_nat0_acl
nat (inside) 1 0.0.0.0 0.0.0.0 0 0
```

The first thing done is to clear the previous settings. The global setting above enables PAT, mapping all interior IP addresses to the one external IP address. NAT 0 enables identity NAT, that is, addresses that will bypass the NAT based on the exemption ACL. NAT 1 states to translate all addresses.

```
static (inside,outside) udp interface 514  192.168.1.2 514  netmask 255.255.255.255 0 0
static (inside,outside) tcp interface smtp 192.168.1.2 smtp netmask 255.255.255.255 0 0
```

If you have servers that require to be accessed from the internet you can use port forwarding to permit access to them. The limitation is that you can only have one server providing the service if you are using PAT.

```
access-group out-in in interface outside
access-group in-out in interface inside
```

Ensure you apply your ACLs to an interface so they are actually used.

```
route outside 0.0.0.0 0.0.0.0 10.2.1.1 1
```

Set up a default route to the internal interface of the router or a default gateway.

```
: pdm locations!!!!
pdm location 192.168.1.4 255.255.255.255 inside
pdm history enable
```

These commands are used by PDM for housekeeping purposes. They are not security related and can be deleted. If deleted you can still use the PDM but may not make configuration changes without the PDM writing these entries.

```
: clear http
: TESTING ONLY uncommented
  http server enable
  http 192.168.1.0 255.255.255.0 inside
```

This enables the http server on the PIX required for PDM access. The next line limits who may access the PDM software. When you have completed the configuration you can use the `clear http` to disable the http server.

```
no snmp-server location
no snmp-server contact
snmp-server community !@#$%smnp-not-used-^&*()
no snmp-server enable traps
```

`SNMP` is not used in environment is therefore disabled. It appears that the PIX always wants to put a community string so we make one that is not easily guessed.

```
floodguard enable
```

This is a default value and it protects against flood attacks.

```
ssh timeout 5
```

Timeout `ssh` sessions after 5 minutes.

```
console timeout 15
```

This command sets up a time out on the console. The default is 0, that is do not time out. A better value may be 15 minutes.

```
clear telnet
clear dhcpd
```

Disable the dhcp server on the PIX since we use only static IP addresses. Also clear the `telnet` server.

```
: this crypto line, below, causes pix to crash!
: Cryptochecksum: 41e32713 f844cdd1 61199081 1b3cb093
Cryptochecksum:eed0cd9711b358c8f0d324f0e7061b31
```

```
: end
```

This checksum is very important to document since it changes anytime the firewall configuration is updated. If you are using a `tftp` server ensure your checksum has no spaces or else it will crash your PIX.

Firewall audit

Planning the audit

The purpose of the audit is to verify our primary firewall performs as we though it should based on our developed rule set. Vulnerability and VPN testing fall outside of the scope of this audit. The audit will be performed as one of the last steps just prior to production implementation. Time will also be planned in case changes are required to be made.

Since this audit is being performed prior to production implementation any improvements and changes will be noted and evaluated for change management review. This is done to be able to rerun the audit once the system has been in production for a time. This is to take into consideration that small changes will occur during the initial production phase.

If this were a production network, we would get prior written permission and would require a lot more planning and communication prior to starting the audit.

Part of our ongoing security practices will be to run periodic audits to catch any changes prior to attackers taking advantage of any weaknesses.

Prior to performing the audit we need to ensure we have the current and final configuration loaded and documented. Documentation includes a printed copy of the configuration.

The audit is divided into sections:

1. testing access to the firewall itself
2. testing available services from the internet
3. testing port forwarding from the internet
4. testing blocked services from the internet
5. testing available services towards the internet
6. testing blocked services towards the internet

The following test scenarios will be used to carry out our audit.

Section 1 – testing access to the firewall itself. This section ensures that the firewall is physical secure from un-authorized persons. This also includes determining what services are available on the firewall itself.

Sections 2 through 6 audit the firewall rule base. The goal is to find ports that are not filtered.

Prior to executing the audit against the running firewall we could take the printed

configuration and run the test cases through manually. This manual test will, hopefully, catch major discrepancies prior to running the actual audit.

The following tools will be used to conduct the audit.

paper, `ping`, `nslookup`, `telnet`, `nmap`, `tcpdump`

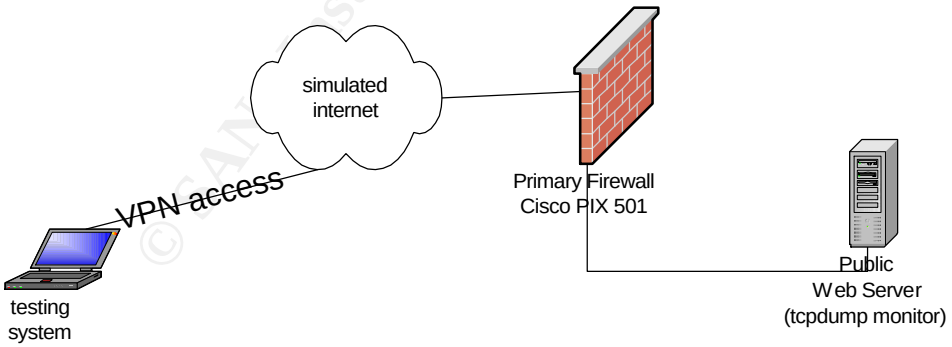
The following estimates, with costing, were submitted to Rune management.

Task description	**Effort**	**Cost ($100/hr)**
Planning	4 hours	400
Desk testing	8 hours	800
Setup	4 hours	400
Test suite 1 execution	2 hours	200
Test suite 2 – 4 execution	8 hours	800
Test suite 5 – 6 execution	4 hours	400
Analysis and recommendations	8 hours	800
Totals	38 hours	3,800

Conducting the audit

The audit would begin by setting up our test environment, shown below, and ensuring we have rebooted the firewall to ensure that the saved configuration is the correct one.

The methodology will be to scan from one side using the `nmap` scanner and examining the firewall logs for proper responses. Also, `tcpdump` is used on the other side to validate traffic was either let in or not. Once one side is completed the scanner and sniffer are reversed and the process starts over.



Testing from the outside

Testing system IP: 10.2.1.1
External firewall IP: 10.2.1.2
Internal firewall IP: 192.168.1.1

Test access to the firewall

```
# telnet 10.2.1.2
Trying 10.2.1.2...
```

PIX syslog

```
Feb 19 00:57:49 192.168.1.1 p_fw %PIX-4-402106: Rec'd packet not an
IPSEC packet. (ip) dest_addr= 10.2.1.2, src_addr= 10.2.1.1, prot= tcp
```

`tcpdump` also confirms no response from the firewall. The above PIX message occurs when you try to access the telnet port on the outside interface without being encrypted.

Test access to public web server

```
# nmap –n –P0 –sS –p80 10.2.1.2
Starting nmap 3.50 ( http://www.insecure.org/nmap/ )

Interesting ports on 10.2.1.2:

PORT   STATE SERVICE

80/tcp open  http

Nmap run completed -- 1 IP address (1 host up) scanned in 0.019 seconds
```

PIX syslog

```
Feb 18 23:56:43 192.168.1.1 p_fw %PIX-6-302013: Built outbound TCP
connection 17 for outside:10.2.1.1/22 (10.2.1.1/22) to
inside:192.168.1.4/1128 (10.2.1.2/1028)

Feb 18 23:57:40 192.168.1.1 p_fw %PIX-6-305011: Built static TCP
translation from inside:192.168.1.2/80 to outside:10.2.1.2/80
```

`tcpdump` also confirms that the packets got through the firewall to the public web server.

Test access for DNS zone transfers

```
# nmap –n –P0 –sS –p53 10.2.1.2
Starting nmap 3.50 ( http://www.insecure.org/nmap/ )

Interesting ports on 10.2.1.2:
```

```
PORT   STATE SERVICE

53/tcp open  domain

Nmap run completed -- 1 IP address (1 host up) scanned in 36.119
seconds
```

PIX syslog

```
Feb 19 00:09:04 192.168.1.1 p_fw %PIX-7-710005: TCP request discarded
from 10.2.1.1/62952 to outside:10.2.1.2/domain
```

`tcpdump` also confirms that no packets got through the firewall.

Test stateful firewall

```
# nmap -n -P0 -sF -p80  10.2.1.2

Starting nmap 3.50 ( http://www.insecure.org/nmap/ ) at 2004-02-19
00:37 EST
Interesting ports on 10.2.1.2:
PORT   STATE SERVICE
80/tcp open  http

Nmap run completed -- 1 IP address (1 host up) scanned in 12.026
seconds
```

PIX syslog

```
Feb 19 00:24:56 192.168.1.1 p_fw %PIX-6-106015: Deny TCP (no
connection) from 10.2.1.1/52522 to 10.2.1.2/80 flags FIN  on interface
outside
```

`tcpdump` also confirms that no packets got through the firewall.

Test access for spoofed addresses

```
# nmap -n -P0 -sS -p80 -S 127.0.0.0 -e eth0 10.2.1.2
Starting nmap 3.50 ( http://www.insecure.org/nmap/ )
Interesting ports on 10.2.1.2:
PORT   STATE    SERVICE
80/tcp filtered http

Nmap run completed -- 1 IP address (1 host up) scanned in 36.134
seconds
```

PIX syslog

```
Feb 19 00:21:30 192.168.1.1 p_fw %PIX-2-106016: Deny IP spoof from
(127.0.0.0) to 10.2.1.2 on interface outside
```

`tcpdump` also confirms that no packets got through the firewall.

Testing from the inside

Testing system IP: 192.168.1.2

Test `telnet` access to the firewall

```
# telnet 192.168.1.1
Trying 10.2.1.2...
telnet: connect to address 192.168.1.1: Connection refused
```

PIX syslog

```
Mar  1 16:02:07 192.168.1.1 p_fw %PIX-3-710003: TCP access denied by ACL from
192.168.1.2/32729 to inside:192.168.1.1/telnet
```

`tcpdump` also confirms no response from the firewall. The above PIX message states the `telnet` connection was refused.

Test `ssh` access to the firewall

```
# ssh -l pix 192.168.1.1
```

PIX syslog

```
Mar  1 15:33:28 192.168.1.1 p_fw %PIX-7-710001: TCP access requested
from 192.168.1.2/37349 to inside:192.168.1.1/ssh
Mar  1 15:33:28 192.168.1.1 p_fw %PIX-7-710002: TCP access permitted
from 192.168.1.2/37349 to inside:192.168.1.1/ssh
Mar  1 15:33:45 192.168.1.1 p_fw %PIX-6-605005: Login permitted from
192.168.1.2/37349 to inside:192.168.1.1/ssh for user "pix"
```

`tcpdump` confirms access to the firewall. The first communication from the administration system will ask you to confirm the hosts RSA1 fingerprint. Subsequent connections do not ask for the confirmation.

Test access to public web server

```
# nmap -n -P0 -sS -p80 2.2.0.8
Starting nmap 3.50 ( http://www.insecure.org/nmap/ )
Interesting ports on 2.2.0.8:
PORT   STATE SERVICE
80/tcp open  http
Nmap run completed -- 1 IP address (1 host up) scanned in 0.019 seconds
```

PIX syslog

```
Mar  1 15:24:57 192.168.1.1 p_fw %PIX-6-305011: Built dynamic TCP translation
from inside:192.168.1.2/16954 to outside:10.2.1.2/1656

Mar  1 15:24:57 192.168.1.1 p_fw %PIX-6-302013: Built outbound TCP connection
721 for outside:2.2.0.8/80 (2.2.0.8/80) to inside:192.168.1.2/16954
(10.2.1.2/1656)

Mar  1 15:24:57 192.168.1.1 p_fw %PIX-5-304001: 192.168.1.2 Accessed URL
2.2.0.8:/
```

`tcpdump` also confirms that the packets got through the firewall to the public web server.

Test access to a public `ftp` server

```
# nmap –n –P0 –sS –p80 2.2.0.8
Starting nmap 3.50 ( http://www.insecure.org/nmap/ )

Interesting ports on 2.2.0.8:

PORT   STATE SERVICE

21/tcp closed  http

Nmap run completed -- 1 IP address (1 host up) scanned in 0.019 seconds
```

PIX syslog

```
Mar  1 15:39:33 192.168.1.1 p_fw %PIX-6-106100: access-list in-out denied tcp
inside/192.168.1.2(58439) ->      outside/2.2.0.8(21) hit-cnt 1 (first hit)
```

`tcpdump` also confirms that the packets did not get through the firewall to the public ftp server.

Test stateful firewall

```
# nmap -n -P0 -sF -p80  192.168.1.1

Starting nmap 3.50 ( http://www.insecure.org/nmap/ )
Interesting ports on 192.168.1.1:
PORT   STATE  SERVICE
80/tcp closed http

Nmap run completed -- 1 IP address (1 host up) scanned in 12.026
seconds
```

PIX syslog

```
Mar  1 15:38:36 192.168.1.1 p_fw %PIX-6-106015: Deny TCP (no
connection) from 192.168.1.2/43786 to 2.2.0.8/80 flags FIN  on
interface inside
```

`tcpdump` also confirms that no packets got through the firewall.

Test access for spoofed addresses

```
# nmap -n -P0 -sS -p80 -S 127.0.0.0 -e ep1 192.168.1.1
Starting nmap 3.50 ( http://www.insecure.org/nmap/ )
Interesting ports on 10.2.1.2:
PORT   STATE    SERVICE
80/tcp filtered http

Nmap run completed -- 1 IP address (1 host up) scanned in 36.134
seconds
```

PIX syslog

```
Mar  1 15:41:50 192.168.1.1 p_fw %PIX-6-110001: No route to 127.0.0.0
from 192.168.1.1
Mar  1 15:41:51 192.168.1.1 p_fw %PIX-7-710005: TCP request discarded
from 127.0.0.0/41453 to inside:192.168.1.1/www
```

`tcpdump` also confirms that no packets got through the firewall but the above PIX message states that the reason could have been that there was no route. The same message was received when we tested another private address range.

Analyzing the results

The audit concludes that the firewall rule base performs as designed from the tests ran. Access to the firewall is limited to `ssh`. Packets allowed out and in pass through the firewall pass. Packets not allowed through are stopped and logged. These results will be documented and will form as the baseline for the firewall. As other audits are conducted their results will be compared to the baseline and any changes will need to be examined.

One of three outstanding issues is regarding the "`no route to`" error message received when testing the private addresses. This message will have to be further researched prior to production implementation. Also, our router should drop these packets given us an extra layer of protection. This would be confirmed will an audit as well.

The second issue is to determine why the `out-in` ACL for DNS replies is not get triggered.

The third issue is to ensure that the `established` keyword is really required on the border router or should another ACL provide coverage.

Since the audit only covered the primary firewall the analysis is focus on these

results. As part of a complete security approach, a complete audit of the security posture needs to be performed prior to implementation and periodically afterwards.

The following recommendations were made to Rune Enterprise management.

- Upgrade router memory.

By upgrading the router memory you can take advantage of additional IOS feature sets including `ssh` version 1 support. This would improve the current `telnet` router access method. Also, install `ntp` on the router. The would allow the router, firewall and IDS to coordinate time for better log analysis.

- VPN device

By separating the VPN functionality from the firewall simplifies the role the firewall plays. Also, separating the two services reduces the load on the firewall for increasing traffic loads.

- Ensure security systems are kept up to date.

This is a very important step since Rune Enterprises does not have a full time security support person. Technical support along with the programmers should subscribe to security mailing lists to keep aware of new vulnerabilities and patches.

- Schedule ongoing Periodic Audits.

This important item is to ensure that the security posture does not degrade as business changes over time.

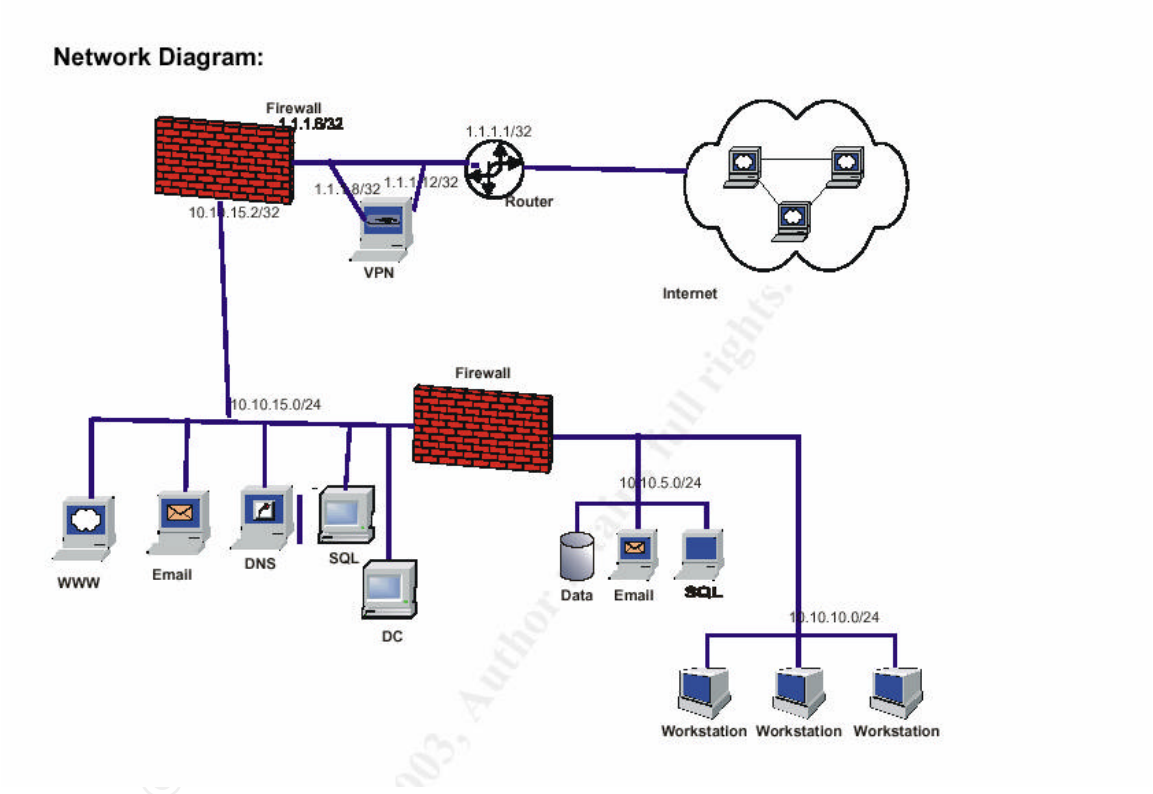
Design under fire

Design selected

The following design was chosen, number 444 Lesa Ludwig[20] submitted on 20th of October, 2003.

The border router is a Cisco 1711 running IOS 12.2 and the primary firewall is IPCops and specialized Linux distribution with an easy to configure firewall based on `iptables`.

Below is a copy of her network design.



The plan

The first step in any attack is reconnaissance, to learn as much as you can prior to any active action. The reconnaissance plan is divided into two parts, technical

and psychological. The technical part is to learn about the network with out being detected prior to an attack. The psychological will be to learn about company details without given away the purpose of the information gathering.

Information regarding GIAC enterprises would be gathered from public information such as news papers, annual reports, public records. Also, competitors, such as, Dark shadows or People's Victorious Fortunes may have information about GIAC's people or their network.

Other approaches could include a visit to their corporate office, if the location is near by, and see if any information can be gathered onsite In addition, once you have a phone number you could call their offices and see what information is leaked through social engineering.

Your reconnaissance should also involve legitimate visits to their web site to determine application versions and operating environments. You can also extrapolate a naming convention on their email addresses if they use real names instead of job functions. You can also purchase a fortune from GIAC and see if any version information is present in any of the communications.

From public records determine the time of year for the company's year end and plan the attack at that time. Thus there may be a chance that all support staff are busy and may miss some of the clues from our attack. Also, you can search for any press releases to assist you in determining names and email address on the corporate users. Other reconnaissance ideas would include passive O/S fingerprinting, war dialing modems and lots of patience.

Once we get to the actual attack it will take consist of three steps: attacking the firewall itself, a denial of service attack and finally a penetration attack.

The attack

The firewall

IPCop does not appears to have any vulnerabilities associated with the program itself[21].

The alternative is to examine the base operating environment for IPCop; Linux. IPCop is based on Linux 2.4 kernel and thus may have the associated vulnerabilities. Researching www.securityfocus.com 's website shows one potential remote vulnerability, bugid 9356. Returning to IPCops website[22] we see that this vulnerability has been corrected in Fix 7 which was released on 14-Jan-04. In addition, another fix, Fix 8 released 20-Feb-04, has already been put out to handle another vulnerability. Vulnerability bugid 9356 requires local access to be able to exploit the bug. There is proof of concept code also available for this exploit.

Since this is the primary firewall we would assume that it has limited access. In addition, being the first line of defense this system would be patched in a short order given the severity of the vulnerability thus negating a possible attack. It thus appears that both IPCop and the underlying Linux layer are well protected from recent vulnerabilities.

Below is a summary of the 9356 vulnerability from the nist.gov website[23].

Vulnerability Name: This reference is to a non-NIST site.	**CAN-2003-0985**
Published before:	1/20/2004
Summary:	The mremap system call (do_mremap) in Linux kernel 2.2, 2.4, and 2.6 does not properly perform bounds checks, which allows local users to cause a denial of service and possibly gain privileges by causing a remapping of a virtual memory area (VMA) to create a zero length VMA.
Severity:	High
Vulnerability type:	Input Validation Error Boundary Condition Error
Exploitable Range:	Local
Loss type:	Security Protection (Gain superuser acess) Availability

Another attack vector could be to examine the rule base configuration file. To do this would require active scanning of the firewall itself that may tip off the support staff.

Distributed Denial of Service (DDoS)

The objective of this attack is to cause a denial of service to GIAC Enterprises. To begin we would compromise 50 or more DSL/cable modem systems. This would be done by employing our favorite virus writer to help us create a custom virus. This method of infection is still very effective[24]. She would modify the W32/MyDoom-A virus for two reasons. The first is to try to bypass virus scanners for enough time to inflect some machines if virus software protected. The second modification is to include the TFN2K DDoS tool.

This tool was selected since it slaves are silent and difficult to spot[25]. Also, the communication between the master and slave are encrypted. Our target systems would be selected from email harvesting SPAM and jokes we have received. This tool consists of a master and many slaves. Various attacks can be launched; TCP, UDP and ICMP packet flooding.

Ways to dampen this type of attack are as follows[26]: be a good net citizen and do egress and ingress filtering, disable un-required broadcast and multicast traffic, implement a incident handling process and protect your systems.

Penetration

The system chosen would be the email server since through our dealings with GIAC we have determined they are running an Exchange 2000 server.

One of the only vectors to get into a network with out breaking defenses is email. Email usually allowed into networks either directly or via relays. The challenge is most are protected by email and virus scanners. The email system makes great target to get into a network since the end points are usually user workstations and possible subject to social engineering.

The following recent vulnerability was found[27] affecting Exchange servers. A proof of concept exploit is also available.

Vulnerability Name: This reference is to a non-NIST site.	CAN-2003-0714
Published before:	11/17/2003
Summary:	The Internet Mail Service in Exchange Server 5.5 and Exchange 2000 allows remote attackers to cause a denial of service (memory exhaustion) by directly connecting to the SMTP service and sending a certain extended verb request, possibly triggering a buffer overflow in Exchange 2000.
Severity:	High
Vulnerability type:	Input Validation Error Boundary Condition Error
Exploitable Range:	Remote
Loss type:	Security Protection (Gain other access) Availability

Would any of our attack activity be detected? According to the design there are

no intrusion detection systems. This does not mean that no malicious traffic would be noticed. `iptables` has some very good logging capacity. Detection would be based on how aggressive we are with our reconnaissance and attack.

The best counter measures are to ensure that you are vigilant with applying patches to all your exposed systems. This requires that companies dedicate staff to keep up to date with announcements and ensure proper testing and change management for applying the patches to their production environments.

Appendix I

PIX configuration file after `clear factory` command issued

```
vax8(config)# conf fac
Begin to apply factory-default configuration:
Clear all configuration
Excuting command: interface ethernet0 auto
Excuting command: interface ethernet1 100full
Excuting command: ip address outside dhcp setroute
.....................
DHCP command failed
Excuting command: ip address inside 192.168.1.1 255.255.255.0
Excuting command: global (outside) 1 interface
outside interface address added to PAT pool
Excuting command: nat (inside) 1 0 0
Excuting command: http server enable
Excuting command: http 192.168.1.0 255.255.255.0 inside
Excuting command: dhcpd address 192.168.1.2-192.168.1.129 inside
Excuting command: dhcpd auto_config
Excuting command: dhcpd enable inside
Excuting command: pdm logging informational
Excuting command: timeout xlate 0:05:00
Factory-default configuation is completed

pixfirewall(config)# sh run
: Saved
:
PIX Version 6.3(1)
interface ethernet0 auto
interface ethernet1 100full
nameif ethernet0 outside security0
nameif ethernet1 inside security100
enable password u0yFmxxxxxgXBm5W encrypted
passwd u0yFmxxxxxXBm5W encrypted
hostname pixfirewall
fixup protocol ftp 21
fixup protocol h323 h225 1720
fixup protocol h323 ras 1718-1719
fixup protocol http 80
fixup protocol ils 389
fixup protocol rsh 514
fixup protocol rtsp 554
fixup protocol sip 5060
fixup protocol sip udp 5060
fixup protocol skinny 2000
fixup protocol smtp 25
fixup protocol sqlnet 1521
names
pager lines 24
mtu outside 1500
mtu inside 1500
ip address outside dhcp setroute
ip address inside 192.168.1.1 255.255.255.0
```

```
ip audit info action alarm
ip audit attack action alarm
pdm logging informational 100
pdm history enable
arp timeout 14400
global (outside) 1 interface
nat (inside) 1 0.0.0.0 0.0.0.0 0 0
timeout xlate 0:05:00
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 TACACS+ protocol tacacs+
aaa-server RADIUS protocol radius
aaa-server LOCAL protocol local
http server enable
http 192.168.1.0 255.255.255.0 inside
no snmp-server location
no snmp-server contact
snmp-server community public
no snmp-server enable traps
floodguard enable
telnet timeout 5
ssh timeout 5
console timeout 0
dhcpd address 192.168.1.2-192.168.1.129 inside
dhcpd lease 3600
dhcpd ping_timeout 750
dhcpd auto_config outside
dhcpd enable inside
terminal width 80
Cryptochecksum:f6b2255cb1d7c984799e9d6aa576500f
: end
pixfirewall(config)#
```

Appendix II

```
: PIX Configuration files - version 1.0
:
: date: 1-mar-04
:

PIX Version 6.3(1)
interface ethernet0 auto
interface ethernet1 100full
nameif ethernet0 outside security0
nameif ethernet1 inside security100
enable password 88ENalbeRyxpa33wd*(&) encrypted
passwd 2TELnet*(&pasWDU encrypted

hostname vax8
domain-name rune.ca
clock timezone EST -5
clock summer-time EDT recurring

fixup protocol http 80
fixup protocol smtp 25

no fixup protocol ftp 21
no fixup protocol h323 h225 1720
no fixup protocol h323 ras 1718-1719
no fixup protocol ils 389
no fixup protocol rsh 514
no fixup protocol rtsp 554
no fixup protocol sip 5060
no fixup protocol sip udp 5060
no fixup protocol skinny 2000
no fixup protocol sqlnet 1521

filter java 80 0 0 0 0
filter activex 80 0 0 0 0
no names

: for VPN to bypass nat id 0
access-list in_out_nat0_acl permit ip any 192.168.5.192   255.255.255.224 log
access-list in_out_nat0_acl permit ip any 192.168.10.192  255.255.255.224 log
access-list in_out_nat0_acl permit ip any 192.168.12.192  255.255.255.224 log
access-list in_out_nat0_acl permit ip any 192.168.15.192  255.255.255.224 log
access-list in_out_nat0_acl permit ip any 192.168.20.192  255.255.255.224 log

: dynamic acls
access-list out_crypto_dyn_20 permit ip any 192.168.5.192 255.255.255.224 log
access-list out_crypto_dyn_20 permit ip any 192.168.10.192 255.255.255.224 log
access-list out_crypto_dyn_20 permit ip any 192.168.12.192 255.255.255.224 log
access-list out_crypto_dyn_20 permit ip any 192.168.15.192 255.255.255.224 log
access-list out_crypto_dyn_20 permit ip any 192.168.20.192 255.255.255.224 log

: VPN peer!
access-list out_20 permit ip any 5.23.16.224 255.255.255.240 log

:
: FIREWALL EXTERNAL INTERFACE
```

```
:
: log since rtr should stop these!

:  access-list out-in deny ip 10.0.0.0 255.0.0.0 any log
access-list out-in deny ip 172.16.0.0 255.240.0.0 any log
access-list out-in deny ip 192.168.0.0 255.255.0.0 any log
access-list out-in deny ip 224.0.0.0 240.0.0.0 any log
access-list out-in deny ip 127.0.0.0 255.0.0.0 any log
access-list out-in deny ip 0.0.0.0 255.0.0.0 any log
access-list out-in deny ip 255.255.255.255 255.255.255.255 any log

: public web and mail servers
access-list out-in permit tcp any host 10.2.1.2 eq 80
access-list out-in permit tcp any host 10.2.1.2 eq 443
access-list out-in permit tcp any host 10.2.1.2 eq 25

: syslog server
access-list out-in permit udp host 10.2.1.1 host 10.2.1.2 eq syslog

: DNS servers!
access-list out-in permit udp host 2.2.0.10 host 10.2.1.2 eq 53
access-list out-in permit udp host 2.2.0.8 host 10.2.1.2 eq 53


:TESTING ONLY
:access-list out-in permit udp any any log
:access-list out-in permit tcp any any log
:access-list out-in permit icmp any any log
:access-list out-in permit ip any any log

: PRODUCTION
access-list out-in deny ip any any log

:
: FIREWALL internal INTERFACE
:

: egress
access-list in-out deny ip 10.0.0.0 255.0.0.0 any log
access-list in-out deny ip 172.16.0.0 255.240.0.0 any log

:  access-list in-out deny ip 192.168.0.0 255.255.0.0 any log

access-list in-out deny ip 224.0.0.0 240.0.0.0 any log
access-list in-out deny ip 127.0.0.0 255.0.0.0 any log
access-list in-out deny ip 0.0.0.0 255.0.0.0 any log
access-list in-out deny ip 255.255.255.255 255.255.255.255 any log


: allow to outside web sites, http, https
access-list in-out permit tcp 192.168.1.0 255.255.255.0 any eq 80
access-list in-out permit tcp 192.168.1.0 255.255.255.0 any eq 443

: only allow mail access to our mail relay
access-list in-out permit tcp 192.168.1.0 255.255.255.0 host
192.168.1.4 eq 25
```

```
: ADMIN stuff ( ssh out )
access-list in-out permit tcp 192.168.1.0 255.255.255.0 any eq 22

: TEMPORARY ADMIN stuff ( TELNET to router out )
: access-list in-out permit tcp 192.168.1.0 255.255.255.0 any eq 23


: DNS servers
access-list out-in permit udp any host 2.2.0.10 eq dns
access-list out-in permit udp any host 2.2.0.8 eq dns

: TESTING ONLY
:access-list in-out permit udp any any log
:access-list in-out permit tcp any any log
:access-list in-out permit ip any any log

:  limited access rules is taken care by the bsd firewall
:  based on vpn pool ips !!!
: access-list in-out deny ip 192.168.1.4 255.255.255.255 192.168.10.192
255.255.255.224 log

: PRODUCTION
access-list in-out deny ip any any log

pager lines 22

logging on
logging buffered warnings
logging trap debugging
logging device-id string p_fw
logging host inside 192.168.1.7

mtu outside 1500
mtu inside 1500
ip address outside 10.2.1.2 255.255.255.252
ip address inside 192.168.1.1 255.255.255.0

ip audit info action alarm
ip audit attack action alarm

ip local pool vpooltest  192.168.5.200-192.168.5.210
ip local pool vpoolexec  192.168.10.200-192.168.10.210
ip local pool vpoolsales 192.168.12.200-192.168.12.210
ip local pool vpoolsup   192.168.15.200-192.168.15.210
arp timeout 14400

:clear global to fac defa reset prior to reconfig
clear global
global (outside) 1 interface

: clear nat for fac default no erro msg
clear nat
nat (inside) 0 access-list in_out_nat0_acl
nat (inside) 1 0.0.0.0 0.0.0.0 0 0

static (inside,outside) udp interface 514 192.168.1.2 514 netmask 255.255.255.255 0 0
static (inside,outside) tcp interface www  192.168.1.2 www  netmask 255.255.255.255 0 0
```

```
static (inside,outside) tcp interface 443  192.168.1.2 443  netmask 255.255.255.255 0 0
static (inside,outside) tcp interface smtp 192.168.1.2 smtp netmask 255.255.255.255 0 0

access-group out-in in interface outside
access-group in-out in interface inside

route outside 0.0.0.0 0.0.0.0 10.2.1.1 1

: nat'd behind 192.168.1.7!!!!
route inside 192.168.4.0 255.255.255.0 192.168.1.7 1

timeout xlate 0:05:00
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

clear aaa-server
: aaa-server TACACS+ protocol tacacs+
: aaa-server RADIUS protocol radius
: aaa-server LOCAL protocol local
: aaa authorization command LOCAL

clear http
no snmp-server location
no snmp-server contact
snmp-server community !@#long***(*#_Q(R%(_V(__####$%smnp-not-used^&*()
no snmp-server enable traps
floodguard enable

: vpn stuff
sysopt connection permit-ipsec

crypto ipsec transform-set t-aes esp-aes-256 esp-sha-hmac
crypto ipsec transform-set t-3des esp-3des esp-sha-hmac

crypto dynamic-map outside_dyn_map 20 match address out_crypto_dyn_20
crypto dynamic-map outside_dyn_map 20 set transform-set t-3des t-aes

crypto map outside_map 20 ipsec-isakmp
crypto map outside_map 20 match address out_20
crypto map outside_map 20 set peer 4.2.2.4
crypto map outside_map 20 set transform-set t-aes
crypto map outside_map 65535 ipsec-isakmp dynamic outside_dyn_map

crypto map outside_map interface outside

isakmp enable outside
isakmp key 123CxYP33RstrongP833wd# address 4.2.2.4 netmask 255.255.255.255 no-
xauth no-config-mode

isakmp policy 10 authentication pre-share
isakmp policy 10 encryption 3des
isakmp policy 10 hash sha
isakmp policy 10 group 2
isakmp policy 10 lifetime 86400
```

```
isakmp policy 20 authentication pre-share
isakmp policy 20 encryption aes-256
isakmp policy 20 hash sha
isakmp policy 20 group 2
isakmp policy 20 lifetime 86400

vpngroup vpnsale2 address-pool vpoolsales
vpngroup vpnsale2 idle-time 900
vpngroup vpnsale2 password Sales1Pa33wd

vpngroup vpnsale1 address-pool vpoolsales
vpngroup vpnsale1 idle-time 900
vpngroup vpnsale1 password Sales1Pa33wd

vpngroup vpnexec address-pool vpoolexec
vpngroup vpnexec idle-time 900
vpngroup vpnexec password exec$$win

vpngroup vpnsup address-pool vpoolsup
vpngroup vpnsup idle-time 900
vpngroup vpnsup password Wh11m333

banner exec Unauthorized use prohibited
clear telnet
:clear ssh
ssh 192.168.1.0 255.255.255.0 inside
: timeout in minutes
ssh timeout 15
console timeout 15
clear dhcpd

terminal width 80
: get PRODUCTION check sum when ready!!!!!
: this crypto line causes pix to crash!!!
: Cryptochecksum: 41e32713 f844cdd1 61199081 1b3cb093
Cryptochecksum:eed0cd9711b358c8f0d324f0e7061b31
: end
```

Appendix III

```
! 1720 Router configuration file - version 1.0
!  1-mar-04
!
version 12.3
no service pad
no service config
no service tcp-small-servers
no service udp-small-servers
no service finger
no service dhcp
no ip finger
no ip classless
no ip source-route
no ip http server
no ip bootp server
no ip domain lookup
no ip subnet-zero
no ip cef
no cdp run
no snmp-server
no boot network
no aaa new-model

no service tcp-keepalives-in
no service tcp-keepalives-out

service timestamps debug uptime
service timestamps log uptime
service password-encryption

banner login # Unauthorized access is prohibited #

hostname vax1
ip domain-name rune.ca

! set up our default route to our ISP router
ip route 0.0.0.0 0.0.0.0 2.2.2.42

boot-start-marker
boot system flash
boot-end-marker
!
!  INTERFACE stuff
interface Ethernet0
 ip address 10.2.1.1 255.255.255.252
 ip access-group 120 in
 half-duplex
 no ip unreachable
 no ip mask-reply
 no ip redirects
 no ip directed-broadcast
 no ip proxy-arp
 no cdp enable
 ip nat inside
```

```
interface FastEthernet0
 ip address 2.2.2.1 255.255.0.0
 ip access-group 110 in
 no ip unreachable
 no ip mask-reply
 no ip redirects
 no ip directed-broadcast
 no ip proxy-arp
 no cdp enable
 speed auto
 ip nat outside
!

ip nat inside source static 10.2.1.2 2.2.2.1
ip nat outside source static 2.2.2.1 10.2.1.1

enable secret 5 $1$strongpH5$enableKqpaswd71
!

memory-size iomem 25

clock timezone EST -5
!
logging buffered 4096 debugging
logging history notifications
logging trap debugging
logging origin-id string c_rtr
logging facility local4
logging source-interface Ethernet0
logging 10.2.1.2

!
! === EXTERNEL interface
!
!
! deny priviate adder
! deny incoming traffic from our internal network (192.168.x.x)
access-list 110 deny ip 192.168.0.0 0.0.255.255 any log
access-list 110 deny ip 172.16.0.0 0.15.255.255 any log
access-list 110 deny ip 10.0.0.0 0.255.255.255 any log
! deny multi cast
access-list 110 deny ip 224.0.0.0 15.255.255.255 any log
! deny loopback
access-list 110 deny ip 127.0.0.0 0.255.255.255 any log
! deny iana   dated: 2004-01-15
! source www.iana.org/assignments/ipv4-address-space
! only denied 0 address since all would be too complex and long
access-list 110 deny ip 0.0.0.0 0.255.255.255 any log
access-list 110 deny ip 255.255.255.255 0.0.0.0 any log

! deny incoming traffic that states being from our firewall-
! router since this is our externel interface and the firewall is
! on our internel interface
access-list 110 deny ip 10.2.1.2 0.0.0.3 any log

!
```

```
! BLACK LISTED hosts and networks
!
! gotomypc.com - 66.151.158.183
access-list 110 deny ip 66.151.158.183 0.0.0.0 any log

!  DENY ACL HERE
! block stuff that could leak out!!!! and dangerous
! windows then unix stuff then others

! windows stuff, netbios, ds, mssql
access-list 110 deny tcp any any range 135 139 log
access-list 110 deny udp any any range 135 139 log
access-list 110 deny tcp any any eq 445 log
access-list 110 deny udp any any eq 445 log
access-list 110 deny tcp any any eq 1443 log
access-list 110 deny udp any any eq 1443 log

!unix stuff
!finger
access-list 110 deny tcp any any eq 79 log
access-list 110 deny udp any any eq 79 log
!tftp
access-list 110 deny udp any any eq 69 log
!rpc
access-list 110 deny tcp any any eq 111 log
access-list 110 deny udp any any eq 111 log

!snmp
access-list 110 deny tcp any any range 161 162 log
access-list 110 deny udp any any range 161 162 log
!r* utils
access-list 110 deny tcp any any range 512 514 log
!nfs
access-list 110 deny tcp any any eq 2049 log
access-list 110 deny udp any any eq 2049 log
! x
access-list 110 deny tcp any any range 6000 6255 log


! deny all icmp except fragmentation required
access-list 110 permit icmp any any 3 4

! TESTING:
! access-list 110 permit icmp any any log

! PRODUCTION:
access-list 110 deny icmp any any log

! ALLOW STUFF follows
!
! allow in stuff
access-list 110 permit tcp any 2.2.2.1 0.0.0.0 eq 80
access-list 110 permit tcp any 2.2.2.1 0.0.0.0 eq 443
access-list 110 permit tcp any 2.2.2.1 0.0.0.0 eq 25

! allow stuff out
access-list 110 permit tcp any 2.2.2.1 0.0.0.0  established log
```

```
! let dns go by since fw will check state
! only dns from our isp's dns servers, eg 2.2.0.8
access-list 110 permit udp 2.2.0.8 0.0.0.0 eq 53 2.2.2.1 0.0.0.0
access-list 110 permit udp 2.2.0.10 0.0.0.0 eq 53 2.2.2.1 0.0.0.0

! vpn check specifics
access-list 110 permit udp any 2.2.2.1 0.0.0.0 eq 500 log
access-list 110 permit esp any 2.2.2.1 0.0.0.0 log

! DEFAULT ACTION

! TESTING ONLY
!access-list 110 permit ip any any log

!PRODUCTION:
access-list 110 deny ip any any log


!
! INTERNEL  interface
!
! deny priviate adder
access-list 120 deny ip 192.168.0.0 0.0.255.255 any log
access-list 120 deny ip 172.16.0.0 0.15.255.255 any log
! access-list 120 deny ip 10.0.0.0 0.255.255.255 any log
! deny multi cast
access-list 120 deny ip 224.0.0.0 15.255.255.255 any log
! deny loopback
access-list 120 deny ip 127.0.0.0 0.255.255.255 any log
! deny iana
access-list 120 deny ip 0.0.0.0 0.255.255.255 any log
access-list 120 deny ip 255.255.255.255 0.0.0.0 any log

! deny traffic that states it is from the outside
access-list 120 deny ip 2.2.2.1 0.0.0.0 any log

! deny all icmp since frag req'd will be handled by outside i/f

! TESTING ONLY
! access-list 120 permit icmp any any log

! PRODUCTION:
access-list 120 deny icmp any any log


!
!  DENY ACL HERE
! block stuff that could leak out!!!! and dangerous
! windows then unix stuff then others

! windows stuff netbios, ds, mssql
access-list 120 deny tcp any any range 135 139 log
access-list 120 deny udp any any range 135 139 log
access-list 120 deny tcp any any eq 445 log
access-list 120 deny udp any any eq 445 log
access-list 120 deny tcp any any eq 1443 log
```

```
access-list 120 deny udp any any eq 1443 log

!unix stuff
!finger
access-list 120 deny tcp any any eq 79 log
access-list 120 deny udp any any eq 79 log
!tftp
access-list 120 deny udp any any eq 69 log
!rpc
access-list 120 deny tcp any any eq 111 log
access-list 120 deny udp any any eq 111 log
!snmp
access-list 120 deny tcp any any range 161 162 log
access-list 120 deny udp any any range 161 162 log
!r* utils
access-list 120 deny tcp any any range 512 514 log
!nfs
access-list 120 deny tcp any any eq 2049 log
access-list 120 deny udp any any eq 2049 log
! x
access-list 120 deny tcp any any range 6000 6255 log

! gotomypc.com - 66.151.158.183
access-list 120 deny ip any 66.151.158.183 0.0.0.0 log


! ALLOW STUFF follows
!
access-list 120 permit tcp 10.2.1.2 0.0.0.0 any eq 80
access-list 120 permit tcp 10.2.1.2 0.0.0.0 any eq 443
access-list 120 permit tcp 10.2.1.2 0.0.0.0 any eq 25
access-list 120 permit tcp 10.2.1.2 0.0.0.0 any eq 22


! TESTING ONLY !!!
access-list 120 permit tcp 10.2.1.2 0.0.0.0 any eq 23

! only dns to our isp's dns servers
access-list 120 permit udp 10.2.1.2 0.0.0.0 2.2.0.10 0.0.0.0 eq 53
access-list 120 permit udp 10.2.1.2 0.0.0.0 2.2.0.8 0.0.0.0 eq 53


! vpn check specifics
access-list 120 permit udp 10.2.1.2 0.0.0.0 any eq 500
access-list 120 permit esp 10.2.1.2 0.0.0.0 any
access-list 120 permit udp 10.2.1.2 0.0.0.0 4.2.2.4 0.0.0.0 eq 500
access-list 120 permit esp 10.2.1.2 0.0.0.0 4.2.2.4 0.0.0.0


! Allow our connections back in
access-list 120 permit tcp any any established log

! DEFAULT ACTION

! PRODUCTION:
 access-list 120 deny ip any any log
```

```
! TESTING
! access-list 120 permit ip any any log
!

! acl for telnet access
access-list 100 permit tcp host 10.2.1.2 23 log
access-list 100 deny ip any any log

line con 0
 session-timeout 5
 exec-timeout 5 0
 password 7 1goodenablepasswd1
 login

! disable aux port no exec
line aux 0
 login local
 no exec

line vty 0 4
 access-class 100 in
 session-timeout 5
 exec-timeout 5 0

 password 7 1goodtelnet1passwdD
 login
!
end
```

References

General

Inside Network Perimeter Security, Stephen Northcutt et al., (New Riders, 2003, ISBN 0-73571-232-8)

www.giac.org/practical/GCFW/Lesa_Ludwig_GCFW.pdf

www.giac.org/practical/GCFW/Atul_Sharma_GCFW.pdf

www.giac.org/practical/GCFW/Darren_Page_GCFW.pdf

www.giac.org/practical/GCFW/Eve_Edelson_GCFW.pdf

www.giac.org/practical/GCFW/Henry_Wong_GCFW.pdf

Brenton, Chris. et al. TCP/IP, SANS Institute. 2003.
Brenton, Chris. et al. Packet Filters, SANS Institute. 2003.
Brenton, Chris. et al. Firewalls, SANS Institute. 2003.
Brenton, Chris. et al. Defense in Depth, SANS Institute. 2003.
Brenton, Chris. et al. VPNS, SANS Institute. 2003.
Brenton, Chris. et al. Network Design and Assessment, SANS Institute. 2003.

Expert Consensus
www.sans.org/top20

Firewall auditing
www.spitzner.net/audit.html

`nmap` Security scanner
www.insecure.org

Router and Firewall commands and more
www.cisco.com

1720 router
http://www.cisco.com/en/US/products/hw/routers/ps221/index.html

PIX 501 Firewall
http://www.cisco.com/en/US/products/hw/vpndevc/ps2030/ps2031/index.html

Endnotes

[1] Dragon Wing, Margaret Weis and Tracy Hickman (Bantam Books, 1990, ISBN 0-553-28639-0)
[2] Idid, page 424
[3] ssh copy really is a scp (secure shell copy)
[4] www.openbsd.org/security.html
[5] Not found.
[6] www.snort.org
[7] The Sans Institute, Securing Linux Step-by-step, version 1.0
[8] www.bastille-linux.org
[9] SANS Parliament Hill, August 2003
[10] www.nsa.gov/snac/cisco/download.htm
[11] VPNs, Chris Brenton (SANS Institue, 2003),, page 3-25
[12] SANS Parliament Hill, August 2003
[13] www.cisco.com/univercd/cc/td/doc/product/iaabu/pix/pix_sw/v_63/cmdref/ab.htm
[14] www.iana.org/assignments/ipv4-address-space, dated 2004-01-15
[15] www.cisco.com/univercd/cc/td/doc/product/iaabu/pix/pix_sw/v_63/cmdref/ab.htm
[16] VPNs, Chris Brenton (SANS Institue, 2003), page 3-25
[17] Secret and Lies, Digital Security in a Networked World, Bruce Schneier (Wiley Publishing, 2004, ISBN-0-471-45380-3) page 90
[18] Ibid, page 94
[19] www.cisco.com/univercd/cc/td/doc/product/iaabu/pix/pix_sw/v_63/cmdref/ab.htm
[20] www.giac.org/pratical/GCFW/Lisa_Ludwig_GCFW.pdf
[21] www.securityfocus.com
[22] www.ipcop.org
[23] icat.nist.gov/icat.cfm?cvename=CAN-2003-0985
[24] www.securityfocus.com/columnists/221, Knock, Knock, Knock
[25] www.riverheadnetworks.com/re/known_ddos_tools.html
[26] www.sans.org/dosstep/roadmap.php#3
[27] www.securityfocus.com/bid/8838