Vincent Berk

ISTS / Dartmouth College
45 Lyme Rd., Suite 200
Hanover, NH 03755
603 646 07476

Track 2: Firewalls, Perimeter Protection and Virtual Private Networks

Assignment part 1

11/22/00

1. Assignment, company description.

The assignment is to produce a security networking architecture implementing as many of the VISA "Ten commandments" as possible. The company is a starting E-Commerce business, expecting to earn $200 million per year in sales of online fortune cookie sayings. The domain name will be: giacfortunes.com.

2. First considerations.

First, before we dive into the technical side of this assignment, I would like to make some important remarks considering the business/management side of this assignment.

a. Starting businesses have very little money.

Since starting a company is risky, the initial funding will probably be minimal. Both banks and the private sector are generally unwilling to invest large sums of money in startups. Furthermore, besides the computer infrastructure, the company will also need office space, employees, furniture, and telephone connections and will face many other initial costs. Starting off with the basic needs seem to be the best (only?) option.

b. Starting businesses need room to expand.

Starting off with the basic needs doesn't mean eliminating all possibilities of expansion. E-businesses tend to expand rapidly depending on the market and the need for bigger and better resources might be necessary and unavoidable. Yet, the financial constants of a starting company probably don't allow for these more expensive resources to be acquired at the very beginning. Thus future growth should be considered in the design. Where scalability is difficult (such as network cabling) installing solid overcapacity should be considered.

c. E-commerce businesses stand or fall depending on their security policy.

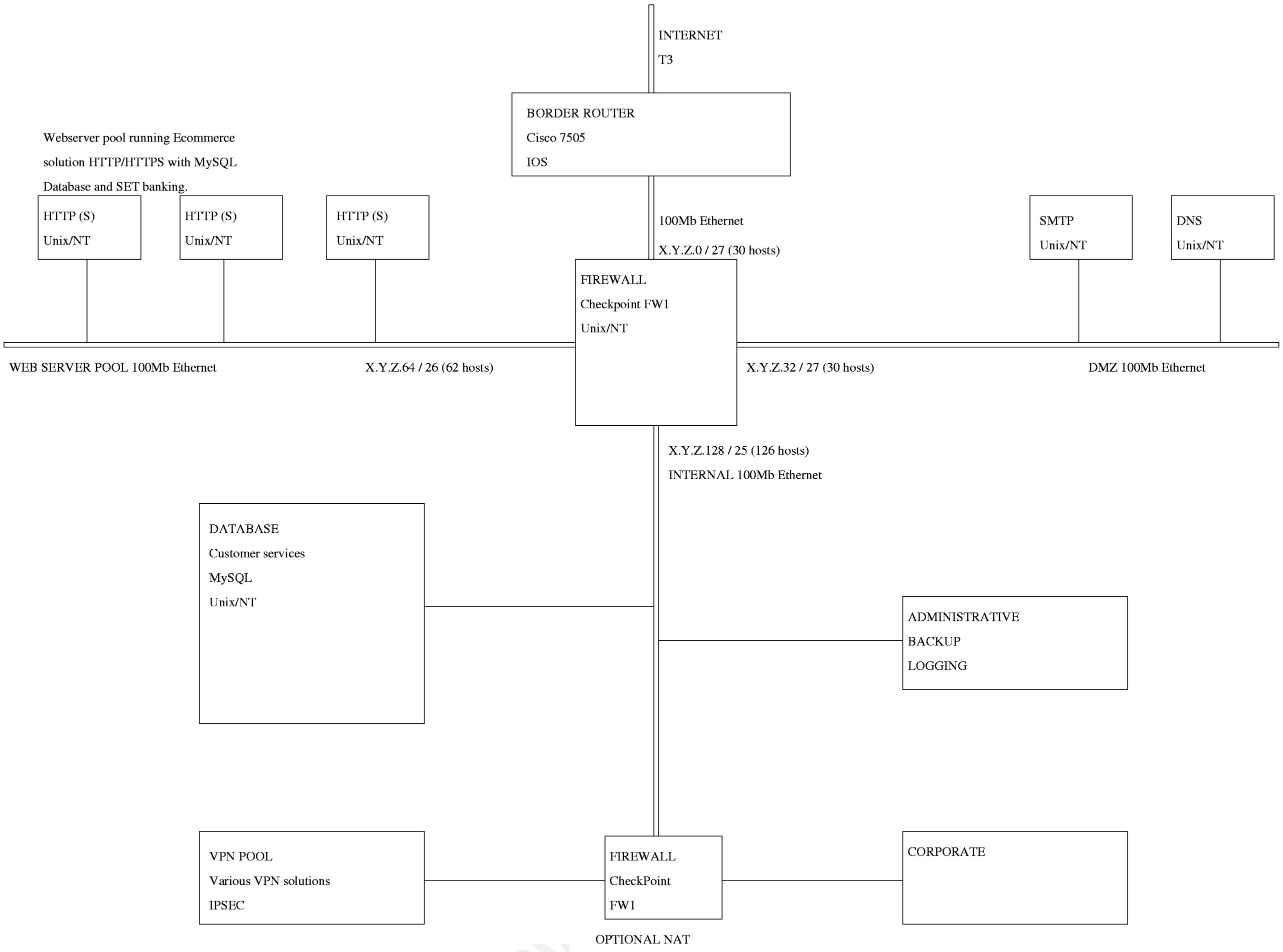
For an E-commerce business the most important means of earning money is through the Internet. This asks for a certain level of trust from both customers and business partners. If this trust is harmed in any way, these parties will most likely reconsider their relationship with the E-commerce business. This could obviously lead to a decrease in sales. Security vulnerabilities should kept to a minimum at all costs in order to keep the chance of an intrusion at an absolute minimum. If the security policy lacks to address important issues or if the security policy is implemented in an incorrect or incomplete way, risks are greatly increased. Care should be taken with the level of skill and field of expertise of the system administrators and the network layout and the security architecture should be clear and kept track of, so that no confusion may arise. The security policy will be further discussed in the next assignment.

d. Every business has customers, suppliers and partners.

There are parties who will require access to the internal network for various reasons. An E-commerce business will probably need a secure connection to a business partner to exchange customer information. There might be mirror sites around the world, which require access to the central database and at least should be kept in mind that such an occasion may arise in the future, as the company grows. Both these examples require secure connections or VPN tunnels over the Internet. This introduces yet another security vulnerability. If the partner network is compromised, the secure tunnel could be used to compromise the company's network. Yet the connection needs to be made. This also is an important point to be taken into consideration.

3. Network Security Architecture.

In this part I will present and walk through the design of the network as to give a clear description of my considerations and the decisions I've made. The design implements multiple layers of defense. I will discuss the layers in the following sections. Multiple defensive layers present a whole series of barriers to break for possible attackers.



Network Overview

a. Bandwidth of incoming connection.

The company aims to make 200 million US dollars a year in fortune cookie sayings. Currently this is all information available regarding network load. For now, let's assume the company will get at most a full T3 connection, roughly 45Mbps. If a much bigger connection with the Internet is needed it might be wise to set up mirror sites instead of having this one site handle all traffic. A single site setup means a single point of failure. Yet for this assignment I won't look into the specifics of setting up multiple mirrors. I will, however, present a method to connect mirrors through VPN connections by setting up a connection to a business partner through a VPN tunnel.

b. Bandwidth of internal connections.

Since the web servers will query the database, the internal network load could be quite considerable. This is why a 100Mbps Ethernet is needed for the internal connections. Furthermore, anything less than 45Mbps would not be able to handle the incoming T3 connection. Finally, 100Mbps Ethernet is most commonly used nowadays, since 10Mbps is generally too slow for average business needs.

c. Border router.

The border router should be able to accept a T3 connection at one side and 100Mbps Ethernet at the other. Furthermore, the border router should be capable of basic filtering on the inbound/outbound traffic, without being a bottleneck. According to bandwidth specifications, a Cisco 7500 series router would be more than sufficient, with the possibility to accept multiple T3 connections for future expansion and filtering processors on each interface (referred to as VIP). Also, the Cisco 7500 series have good expansion possibilities and support both standard Access Control Lists (filtering based on source IP address only) and extended ACL's (filtering based on source and destination IP addresses and port numbers.) This means some attacks (e.g. address spoofing) can be stopped before the packets even enter the network. The border router is the first line of defense.

d. Network segmentation.

The network is a full Class C, using VLSM; Variable Length Subnet Masking. The network address is X.Y.Z.0, thus blanking the first three octets since those are always the same. Using VLSM saves address space, which is very sparse nowadays. Normal subnet masking would give us four sections of 62 hosts maximum each, which could be insufficient for the internal network at a certain point in the future.

Between the router and the firewall:	X.Y.Z.0 / 27 30 hosts maximum
The DMZ segment:	X.Y.Z.32 / 27 30 hosts maximum
The web server segment:	X.Y.Z.64 / 26 62 hosts maximum
Behind the firewall, internal network:	X.Y.Z.128 / 25 126 hosts maximum

So, the network address of the web server pool would be: X.Y.Z.64 and the subnet mask would be 255.255.255.192.

e. OS Platform.

In my view the installed OS should be the one which the operator or administrator is most comfortable and experienced with. This results in better manageability and security of the network. This approach is very reasonable nowadays, since practically all servers are available for multiple platforms. Yet, either Unix (Linux) or NT should be used to keep good track of users and their actions (e.g. when an administrator changes the firewall rulebase).

f. Firewall.

The firewall should be able to handle the bandwidth of the four connected segments. Firewall 1 has been benchmarked on very demanding situations and has proven to be very well capable of sustaining high throughput rates, as much as 280Mbps on a dual CPU system with 4 networking interfaces. It should be considered a very high priority to acquire a very powerful machine for the firewall, since it will probably be the most demanding task of the whole network. Further considerations where ease of use and maintainability, multiple platforms supported and stateful packet filtering. Firewall 1 has a nice GUI, which enables administrators to quickly and efficiently oversee the rulebase and add or remove rules. Furthermore, since it does stateful packet filtering, no rules have to be added to allow for return packets to the higher port numbers (> 1024). Also, with stateful filtering, attacks aimed at exploiting vulnerabilities due to invalid packet sequences are blocked (e.g. after a SYN, we expect a SYN/ACK back,...). More about this is provided in assignment 3, where the security architecture will be audited. As an added feature, Firewall 1 supports VPN1 encryption/decryption. In specific cases this might be of particular use, since FW1/VPN1 are the most commonly used firewall/VPN products of today. More about this is included in the VPN section. (Note: the firewall might be overloaded when VPN encryption/decryption is done too.) The firewall is the second and most important layer of defense.

g. Demilitarized Zone.

The DMZ is located on a separate segment of the firewall. Both the mail server and the two DNS servers are located there. These services are not located on the internal network since the outside world should be able to connect to them. Yet I do not place them outside the firewall since connections to them should be tightly controlled and closely monitored. It is not advisable to have the outside world connect to any machine inside the internal network.

h. Web server pool.

This is where the actual E-commerce work takes place. Since we are dealing with customer information and credit card numbers, the web servers will be running an E-commerce solution supporting HTTPS (Secure Socket Layer). This allows customers to safely connect to the companies web servers through a secure and encrypted connection, verified by the companies' certificates (Verisign provides certification services). For the customer information to be stored, orders to be placed and products to be retrieved, the web servers need to be able to connect to a database server. This database is located inside the internal network. A multi-platform E-commerce solution could be (and has been proved to work very well): Apache or IIS web server with Allaire Coldfusion and the BossCommerce package, which supports HTTPS and is compatible with various databases (including: MySQL, Oracle 8, Microsoft SQL 7, IBM DB2 and MS Access). Case studies have shown that this is a fast and reliable solution for a low price. Furthermore, this combination supports various ways of handling the financial transactions (one of which is SET - Secure Electronic Transmission, which is widely deployed nowadays). For financial transactions a SET connection is made between the web server pool and the financial institution. Since the web server pool is so important to this company, I decided to create a separate

firewall segment for it. If the web servers were in the DMZ with the mail and the DNS servers, a compromise of the latter would mean a severe security threat to the web servers, and subsequently the database. It is known that both Bind (DNS) and Sendmail (Mail) have had very many security vulnerabilities (and subsequently many break-ins). If either of these servers gets compromised, there is no protection for the web servers if they are on the same physical network segment. If a web server gets compromised, the database would be in severe danger of theft and/or modification of customer data. In my design, in case either the mail or the DNS server gets compromised, the web servers are still behind the firewall, thus adding an extra layer of protection. Furthermore, Firewall 1 provides load balancing over a group of servers (called logical server), so the web server pool will act as one big server with the firewall distributing the connections over the various physical servers.
The logical web address will be: www.giacfortunes.com.

i. Database.

The database is located inside the internal network. This is because all the extremely sensitive customer data is stored there. For this company it will be enough to start with a MySQL server on a fast machine with lots of memory and disk space. MySQL has proven to be very reliable even with huge databases of 50 million records or more. If in the future there is any need for a costly parallel database (like Oracle 8) the E-commerce package also supports this.

j. Administrative network.

The administrative consoles, the backup server and the log server are directly connected to the main internal network. This way logging and backups can be done in a rapid and reliable fashion.

k. Internal firewall.

This is an internal firewall to restrict (parts of) the corporate network from connecting to the sensitive data on the main internal network. Furthermore, this firewall takes care of restricting access from the business partner coming in over the VPN channel. Only several machines from the corporate network and several machines from the partners network are allowed to access the customer database.

l. VPN network

This is an optional network protected by a firewall. A Virtual Private Network is needed for connections to partners or mirror sites, so machines are needed to encrypt/decrypt the sessions. I propose this design because of various reasons. First of all, it is good to have some form of protection between the internal network and the end of the VPN tunnel (for we tunnel right through the main firewall). If a partner or mirror site gets compromised, we still have the internal firewall to protect our internal network from being compromised immediately. Secondly, for the sake of performance I don't put the VPN pool on a fifth segment of the main firewall, since it will be busy enough as it is right now (let alone using the main firewall to decrypt Checkpoint's VPN1 traffic). Furthermore, different partners may have different VPN solutions so a machine can be put up for every connection. Finally, the internal firewall can do Network Address Translation in case that's necessary (if, for example, corporate runs out of IP addresses). There are multiple ways of setting up a VPN connection. PPTP - Point to Point Tunneling Protocol is a protocol on top of IP, TCP port 1723. For the purpose of this assignment, there is a business partner who has access to the database (MySQL) through an IPSEC VPN tunnel. IPSEC is a collection of protocols to allow for authentication and encryption of traffic and is most commonly used nowadays. The company's corporate network may initiate connections to the partners network too (e.g. to access their customer database).

An important point should be noted here. The VPN tunnel creates a pathway to the partner machines. This means that packets destined for the partner machines should be routed through the VPN tunnel. What this means is that all packets for this partner network should have get routed to the VPN tunnel host at our local site. On the local network the VPN encryption/decryption host will be set up as a router for the business partners network.

m. Corporate

The machines required for corporate purposes. If there is a shortage of IP addresses, the internal firewall can do Network Address Translation for the corporate network (Checkpoint Firewall 1 is capable of doing NAT).

4. Final considerations.

a. Reverse proxies.

A reverse proxy works the same as a regular proxy server, except that it works for inbound connections. Instead of directly connecting to the web server, the customers connect to the reverse proxy, which in its turn connects to the actual web server behind the firewall. The reason for not using a reverse proxy server is that, especially with SSL, it creates a lot more network traffic through the firewall to the actual web servers. (MySQL queries are a lot smaller than all the traffic going through SSL.) Since the firewall is already heavily loaded, it is a better solution to avoid this extra layer of complexity.

b. Connection Case Study.

Assume a customer on the Internet wants to purchase a personal fortune cookie. They first connect to the website (www.giacfortunes.com) using HTTP (of course, to resolve the address, the DNS server is queried first). The connection gets filtered by the border router and verified by the main firewall. The user chooses to log in so the user will be linked to a HTTPS page, which will be automatically loaded by his/her browser. The same filtering is done as with the HTTP request. After entering the username and the password, the web server queries the database, using MySQL, to validate the login. The main firewall has to approve of this connection.

The user decides to make a purchase. Again the database is queried. This time the web server retrieves payment information and the actual fortunes. Next, the web server makes a SET encrypted connection to the financial service provider, out on the Internet. The firewall checks this connection. Finally, to finalize the purchase, a receipt is emailed to the user. The web server connects to the mail server using SMTP to send the message. (The mail server has to query the DNS server to resolve the destination mail server.)

Vincent Berk

ISTS / Dartmouth College
45 Lyme Rd., Suite 200
Hanover, NH 03755
603 646 07476

Track 2: Firewalls, Perimeter Protection and Virtual Private Networks

Assignment part 2

11/22/00

1. SANS Assignment 2: security policy.

The assignment is to design a security policy based on the SANS top-ten filtering rules with the focus on additional filtering. A manual is to be written on how to implement the policy in the network design of assignment 1. Focus of this will be on the main firewall.

First I will give the security policy to be implemented. Subsequently I will give the manual on how to implement the rules on the network, focusing on the main firewall. The manual will also provide test methods for the implemented rulebase and an extensive description of the rules.

At the end of this document there is a copy of the network graph.

2. Security Policy

The security policy is divided in two parts: a general part and a specific part. A more extensive description of each specific rule is given in the next part of this assignment, the implementation manual. For the remainder of this assignment, when there is the option of using either Unix of Windows NT/2000, the OS of choice will be Unix. Examples will be based on Unix operating systems unless specifically mentioned otherwise.

a. General rules

1. **Patches and Updates.** New software updates and security patches will always be installed as soon as possible in order to keep vulnerabilities to a minimum. System administration is fully responsible for keeping track of this.
2. **Logging.** All changes, installations and modifications will be logged by the system administration to ensure no later confusion.
3. **Authentication and Tracking.** Each system administrator has his/her own unique user ID to track all administrative actions, done by using either Windows NT/2000 or Unix on all systems. Also, all employees have unique user ID's, to track actions and access to data.
4. **Validation and Testing.** All changes, updates, patches, installations and modifications will be fully tested for functionality or flaws. All tests will be logged by system administration.
5. **Accessibility Restrictions.** Data and services will only be accessible by machines with specific requirement. To this extend filtering routers and firewalls will all be configured with a 'Default Deny' rule. Access is only granted when absolutely required to make sure no 'holes' is accidentally left opened. 'Default Deny' rule means that all the packets from any source to any destination will be denied by default. This rule is usually placed as the last rule with least priority in the firewall, after all other specific rules have been applied.

b. Specific rules

1. As a baseline policy, the SANS top-ten is taken, found at: http://www.sans.org/giactc/gcfw.htm. All filtering devices will have the 'Default Deny' rule. Furthermore, they block spoofed address (both inbound and outbound), private addresses and source-routed packets.
2. All ICMP requests are blocked to avoid ping sweeps to succeed (ping sweep: sending ICMP echo requests to a full network to see which hosts are up and reply.) Also, all RIP packets are denied since there is no need to dynamically update routes.
3. Passwords are administrator specific, different for each host and enforced strong (8 characters or more, combination should contain both capitals and numbers, no dictionary words allowed.) This is to minimize the change of passwords being cracked by brute force password guessing.
4. All unneeded services will be removed from each server (this includes removing the executables). Intrusion warnings are sent as an email to the administrator.

- Connections to and from the Border Router.
  The Border Router accepts telnet (TCP/23) connections only from the administrative network. This is to allow for administration. Connection attempts from the Internet will trigger an intrusion warning. The Border Router is allowed to make syslog (UDP/514) connections to the logserver.

- Connections to and from the Mail server.
  The Mail server accepts SMTP (TCP/25) connections from the Internet, the web server pool, the administrative network and the corporate network. This to allow messages to be sent. Furthermore, the mail server accepts POP3 (TCP/110) connections from the administrative network and the corporate network. The Mail server accepts SSH (TCP/22) connections from the administrative machines to allow for administration and backup. The Mail server is allowed to make syslog (UDP/514) connections to the logserver and DNS requests (UDP/53) to the DNS server. All other connections made by the Mail server will trigger an intrusion warning. Message relaying is disabled to avoid the server being used as a SPAM relay center.

- Connections to and from the DNS servers.
  The DNS servers accept DNS requests (UDP/53) from the Internet, the web server pool, the administrative network, the corporate network and the Mail server. This is to allow for DNS requests. Furthermore, the DNS server accepts SSH (TCP/22) connections from the administrative machines to allow for administration and backup. The DNS servers are allowed to make all DNS connections (TCP/53 and UDP/53) to the Internet. This to allow for DNS requests and Zone transfers. Zone transfers between the primary and secondary DNS are allowed (TCP/53 and UDP/53). The DNS servers are allowed to make syslog (UDP/514) connections to the logserver. All other connections made by the DNS servers will trigger an intrusion warning.

Zone transfers are a risk since they allow attackers to quickly gather a lot of sensitive information about the internal network layout (all domain information gets transferred). Enabling them to quickly pick targets for their attacks.

- Connections to and from the web server pool.
  The web server pool accepts HTTP (TCP/80), HTTPS (TCP/443) from the Internet, the administrative network and the corporate network. Furthermore, the web server pool accepts SSH (TCP/22) connections from the administrative machines to allow for administration and backup. The web server pool is allowed to make MySQL (TCP/3306) connections to the database, DNS requests (UDP/53) to the DNS server, SMTP (TCP/25) connections to the Mail server and syslog (UDP/514) connections to the syslog server. Finally, the web server pool is allowed to make SET (TCP/257 and UDP/257) connections to the Internet, for banking services.

- Connections to and from the customer database server.
  The database server accepts MySQL (TCP/3306) connections from the web server pool, the VPN pool (specific business partner machines), specific machines from the corporate network and the administrative machines. Also, the database server accepts SSH (TCP/22) connections from the administrative machines to allow for administration and backup. The database server is allowed to make syslog (UDP/514) connections to the syslog-server. All other connections made by the database server will trigger an intrusion warning.
  Connections to the database server require an account on this server. There are multiple access levels: read-only and read-write. Obviously, the web servers have read-write access, since they accept orders and other customer data. Users in the corporate machine pool or users at the site of the business partner might only have read access. Read and write access is only granted on a need basis. User accounts for the database server will be created by the administrators.

- Connections to and from the VPN pool.
  The VPN pool currently only contains one machine. This machine is allowed to receive and transmit using the IPSEC protocol group with the partner host, on the Internet. Furthermore several specific machines from the business partner are allowed to initiate a MySQL (TCP/3306) connection with the customer database. Finally this VPN host itself is allowed to make syslog (TCP/514) connections with the syslog server and allowed to accept SSH (TCP/22) connections from the administrative network.
  There are only a few machines of the business partner, which get routed through the VPN tunnel. The VPN machine acts as a router for those machines. The DNS and mail servers of the business partner will still be connected to using a normal Internet connection (routed through the border router).

- Connections to and from the corporate network.
  The corporate network does not accept any connections at all. Several specific hosts from the corporate network are allowed to make MySQL (TCP/3306) connections to the database. Furthermore, the corporate network is allowed to make HTTP (TCP/80) and HTTPS (TCP/443) connections to the web server pool, DNS (UDP/53) connections to

the DNS server, SMPT (TCP/25) and POP3 (TCP/110) connections to the Mail server, syslog (UDP/514) connections to the syslog server and all connections to the business partners machines, through the VPN tunnel. Finally, the corporate network is allowed to make connections to the Internet.

- Connections to and from the administrative network.
  The administrative network is allowed to make HTTP (TCP/80), HTTPS (TCP/443) and SSH (TCP/22) connections to the web server pool. SSH is used for administration and backup. Furthermore, the administrative hosts are allowed to (UDP/53 and TCP/53) and SSH (TCP/22) connections to the DNS server, SMTP (TCP/25), POP3 (TCP/110) and SSH (TCP/22) connections to the Mail server, MySQL (TCP/3306) and SSH (TCP/22) connections to the database server, SSH (TCP/22) connections to the VPN host, syslog (UDP/514) and SSH (TCP/22) connections to the syslog server and SSH (TCP/22) connections to the backup server. Also, the administrative network is allowed to make connections to the Internet. The syslog server in the administrative machine pool accepts syslog (UDP/514) packets from the web server pool, the Mail server, the DNS server, the Border Router, the Firewalls, the Database, the VPN encryption/decryption host and the backup server.

- Connections to and from Firewall machines.
  The Firewall machines do not accept any connections at all. All firewall administration is done through the systems local console. The Firewall machines are allowed to have syslog (UDP/514) communication with the syslog server. Also, the main firewall blocks incoming ICMP requests as well as the RIP protocol. This is to avoid information gathering through ping sweeps.

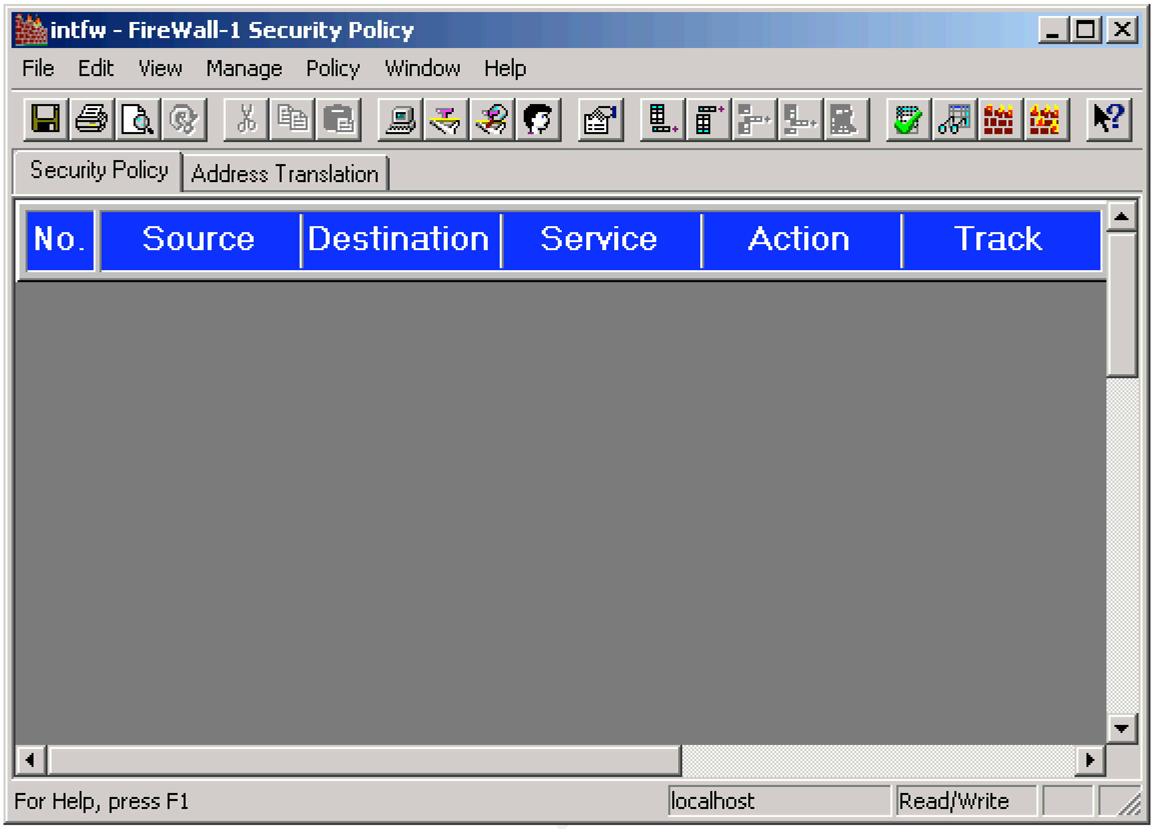
3. Implementation manual

This manual is split up in three parts, the Main Firewall, the Border Router and the Internal Firewall, with the first part being the most important part. For each part I will give a detailed description of how the device is configured. Then I will walk through the security policy and explain which parts should or cannot be implemented in this section.

a. Configuring a Checkpoint Firewall 1

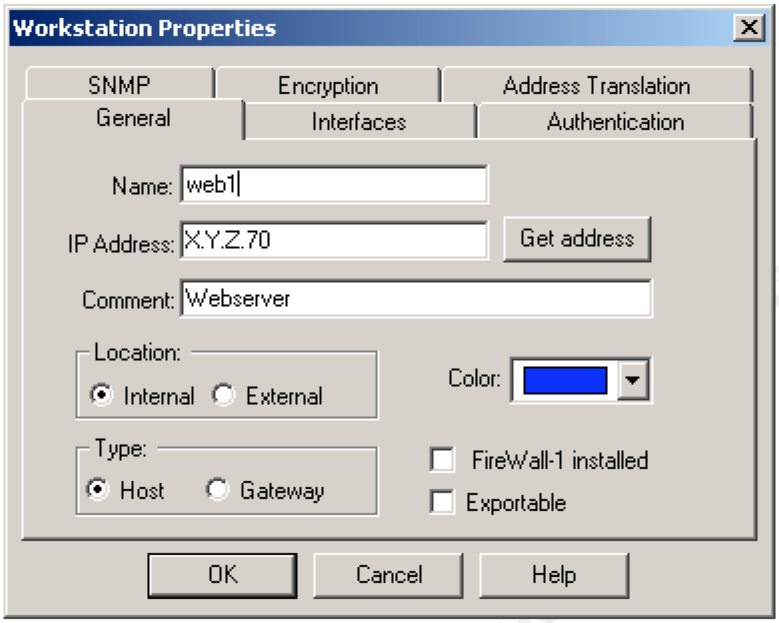
Firewall 1 has a very nice GUI to edit the firewall rulebase. This policy editor allows network objects to be created upon which the rulebase can be applied. Network objects can be anything from a router to workstation, from an address range to a logical server. There are many books and manuals on this so I'll keep it very short with some clear pictures.

Blocking ICMP requests and RIP packets is done in the properties configuration window. Also, the email address of the administrator and the Alert method can be set here. Finally, it is important to block control connections to the firewalls in this configuration window.



This is the Firewall 1 policy editor.

A workstation is a normal network host. This could also be a server, as shown in the example below. In case of servers it is advisable to create a workstation object for every server.



Name: web1

IP Address: X.Y.Z.70

Comment: Webserver

Location: ● Internal ○ External

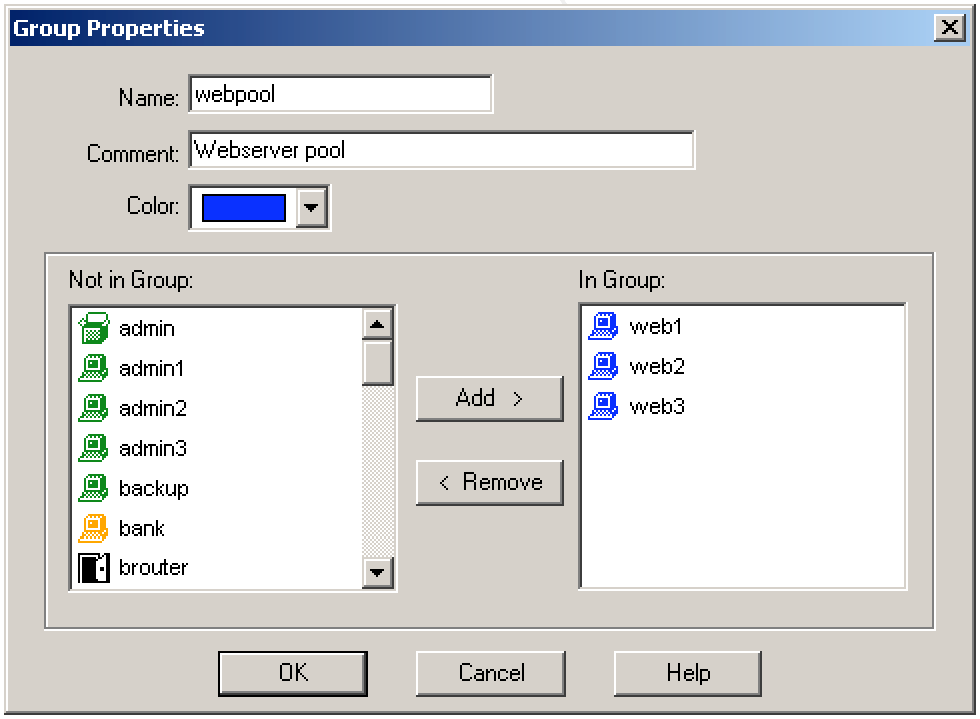
Color: 🟦

Type: ● Host ○ Gateway

□ FireWall-1 installed

□ Exportable

A group is created out of a collection of other network objects. Anything can be grouped together:



Name: webpool

Comment: Webserver pool

Color: 🟦

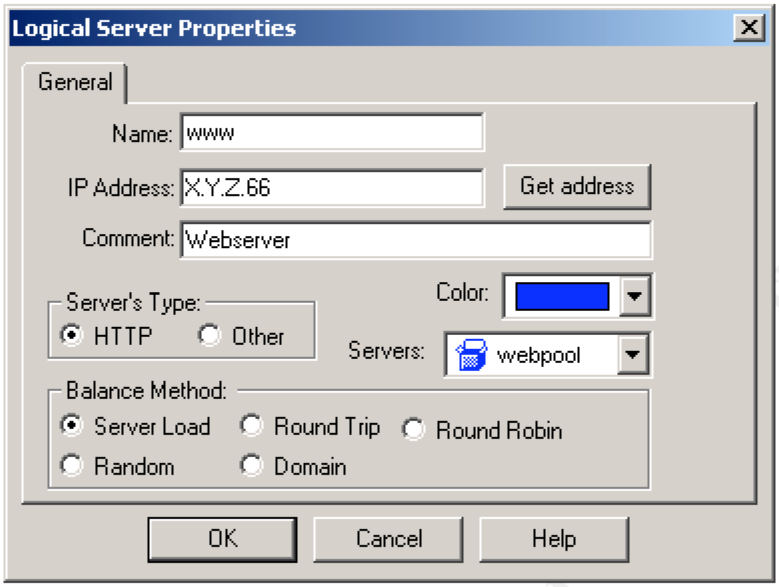
Not in Group:

- admin
- admin1
- admin2
- admin3
- backup
- bank
- brouter

In Group:

- web1
- web2
- web3

A logical server is a group of servers, which are providing the same services, addressed as one bin logical server. Firewall 1 performs load balancing over the servers and make them appear as one physical machine:



Name: www

IP Address: X.Y.Z.66

Comment: Webserver

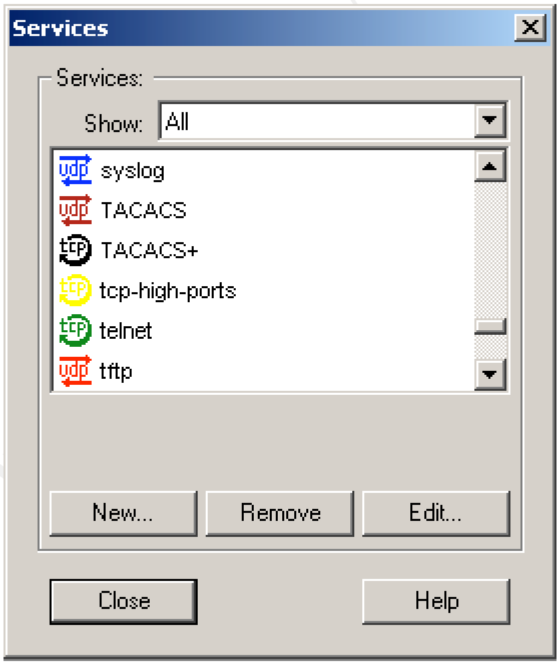
Color: 🟦

Server's Type: ● HTTP ○ Other

Servers: webpool

Balance Method: ● Server Load ○ Round Trip ○ Round Robin ○ Random ○ Domain

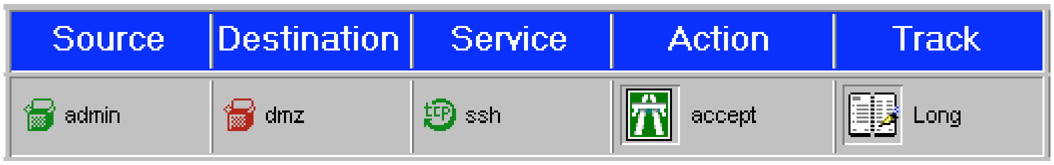
The services menu allows us to add services (protocols) and group them together:



Show: All

- syslog
- TACACS
- TACACS+
- tcp-high-ports
- telnet
- tftp

Each rule in the rulebase has a source network object, a destination network object, a service and an action. When the first three fields match, the action is performed.



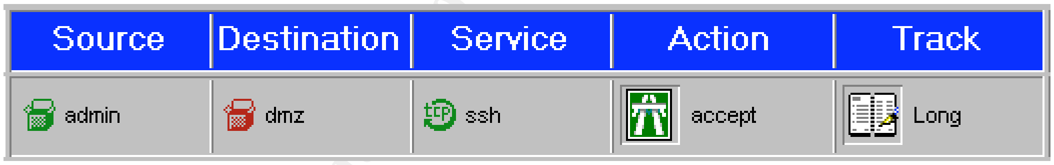
Source	Destination	Service	Action	Track
admin	dmz	ssh	accept	Long

So for this example, if a packet comes in from an admin machine and is destined for a DMZ machine, using protocol SSH (TCP/IP port 22) the action will be executed. Accept means the packet will be let through. Drop is another possibility (there are more possible actions but they won’t be used in this assignment. For a more accurate description see the Firewall 1 manual). Track contains additional action information. It can either be empty or the type of logging to be done. Long means that each connection, matched by this rule, will be logged in the long format. Another option is Alert. Alert will use a specified method to warn the administrator (e.g. through sending an email).

b. Testing and debugging

Each modification should be tested and proven to work. In the beginning I will be very explicit about how to test. Later on, testing will be obvious. I will give a very elaborate example of how to test a new firewall rule. In this I will explain what Checkpoints Firewall 1 rules look like and how they work. Then I will explain how Firewall 1 logfiles work. Finally I will show some useful tools for sniffing network traffic and making network connections.

The firewall rule:



Source	Destination	Service	Action	Track
admin	dmz	ssh	accept	Long

This rule means that the admin group of computers is allowed to initiate a SSH connection to the machines in the DMZ group. Obviously, to test if this rule works, an SSH connection should be made from a machine in the admin group to a machine in the DMZ group:

```
    root@admin1# ssh dns1
```

With dns1 being a machine in the DMZ group. This connection should succeed. If it fails we go debugging, which I will talk about later on. But this is obviously not the only thing we should test. There are two other cases to check:

1. Are we able to reach other (closed) ports on the destination host?
2. Are unauthorized machines able to reach the destination host?

The first case is tested by trying to connect to another service on the destination host, from the admin group. The second case is tested by trying to connect to both SSH and some other services on the destination host, from an unauthorized host, say corporate machine corp4. Obviously all those tests should fail:

```
    root@admin1# telnet dns1
    root@corp4# ssh dns1
```

```
root@corp4# ftp dns1
```

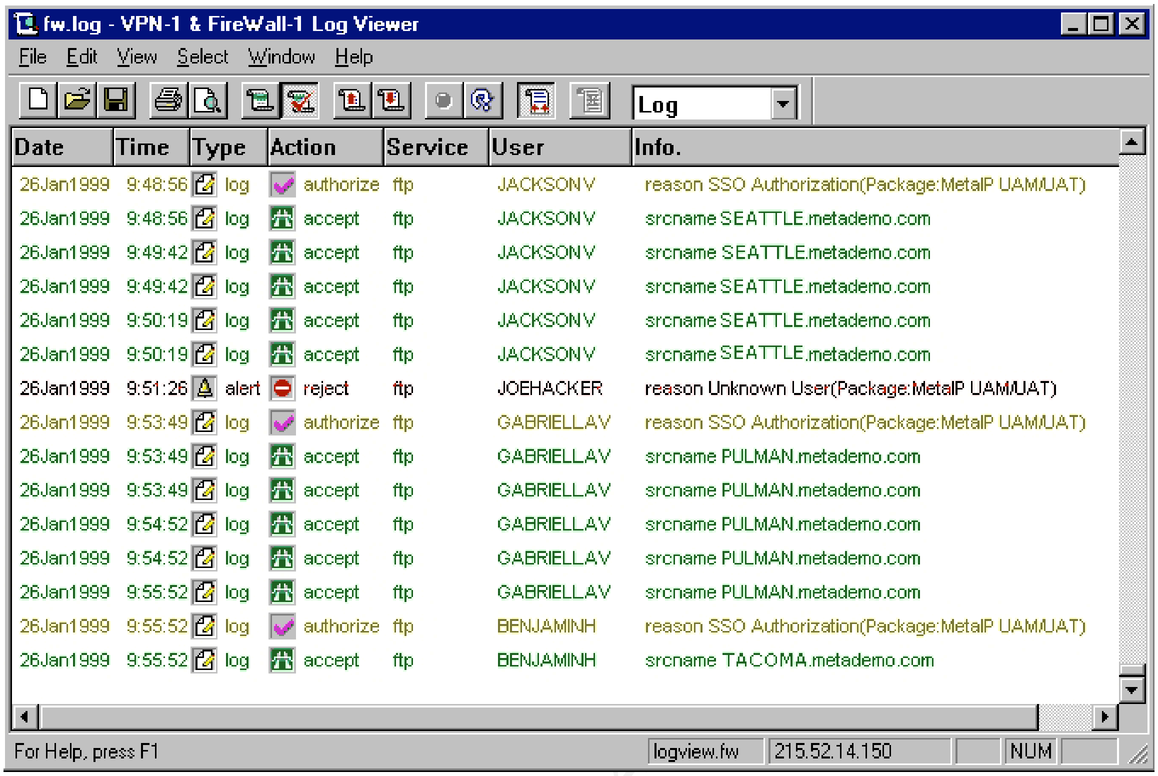
If any of those succeeded, while both telnet and ftp are blocked services for the dns1 host, there is obviously something wrong and the rulebase needs to be looked over. Obviously we can't test for all open services like this, we need a tool. This tool is nmap and has both unix and windows versions. nmap is able to scan for a range of port numbers (TCP as well as UDP) or just a whole port range. How to use nmap, several examples (see nmap documentation for all options):

```
nmap [options] host
nmap dns1                     will give us all reachable TCP services on dns1
nmap -sU dns1                 will give us all reachable UDP services on dns1
nmap -p1-1024 dns1            will scan TCP ports 1 to 1024 on dns1
```

```
root@admin1# nmap dns1

Starting nmap V. 2.12 by Fyodor (fyodor@dhp.com, www.insecure.org/nmap/)
Interesting ports on dns1 (X.Y.Z.35):
Port    State    Protocol  Service
22      open     tcp       ssh

Nmap run completed -- 1 IP address (1 host up) scanned in 0 seconds

root@admin1#
```

Above is a transcript of an nmap session from admin1 to dns1 scanning for known ports. Only port 22/TCP (SSH) showed up. That means that the TCP filter is working. If multiple ports show up unintended, something might be wrong with the firewall rulebase. This tool can also be run from unauthorized hosts, which obviously should give no open ports.

So how do we know what the firewall did? Well, the firewall produces logfiles to show what's been going on. Here is an example log file in the log viewer:


fw.log - VPN-1 & FireWall-1 Log Viewer
File Edit View Select Window Help
Log

Date	Time	Type	Action	Service	User	Info.
26Jan1999	9:48:56	log	authorize	ftp	JACKSONV	reason SSO Authorization(Package:MetaIP UAM/UAT)
26Jan1999	9:48:56	log	accept	ftp	JACKSONV	srcname SEATTLE.metademo.com
26Jan1999	9:49:42	log	accept	ftp	JACKSONV	srcname SEATTLE.metademo.com
26Jan1999	9:49:42	log	accept	ftp	JACKSONV	srcname SEATTLE.metademo.com
26Jan1999	9:50:19	log	accept	ftp	JACKSONV	srcname SEATTLE.metademo.com
26Jan1999	9:50:19	log	accept	ftp	JACKSONV	srcname SEATTLE.metademo.com
26Jan1999	9:51:26	alert	reject	ftp	JOEHACKER	reason Unknown User(Package:MetaIP UAM/UAT)
26Jan1999	9:53:49	log	authorize	ftp	GABRIELLAV	reason SSO Authorization(Package:MetaIP UAM/UAT)
26Jan1999	9:53:49	log	accept	ftp	GABRIELLAV	srcname PULMAN.metademo.com
26Jan1999	9:53:49	log	accept	ftp	GABRIELLAV	srcname PULMAN.metademo.com
26Jan1999	9:54:52	log	accept	ftp	GABRIELLAV	srcname PULMAN.metademo.com
26Jan1999	9:54:52	log	accept	ftp	GABRIELLAV	srcname PULMAN.metademo.com
26Jan1999	9:55:52	log	accept	ftp	GABRIELLAV	srcname PULMAN.metademo.com
26Jan1999	9:55:52	log	authorize	ftp	BENJAMINH	reason SSO Authorization(Package:MetaIP UAM/UAT)
26Jan1999	9:55:52	log	accept	ftp	BENJAMINH	srcname TACOMA.metademo.com

In general log files give us a timestamp (date and time) a source address, a destination address, the service type and the reason why this line is logged. The logfile in the picture shows several ftp connections being made to an ftp-server. Most connections get accepted, several get authorized (username and password are entered correctly) and one is rejected (the red line). Note that the 'track' field from the policy editor shows up in the 'type' field of the log viewer. The reason for this one connection to be rejected is that authorization failed: the username is unknown.

What if there seems to be a problem but the logfiles don't give us what we want to know? In that case we might have to check what packets are physically on the wire. A tool which is excellent in detecting what is going on is tcpdump. tcpdump listens on the network device and prints a summary of every packet that flies by. Very important here is that tcpdump can actually receive the packets, so be careful not to put the detection host on a normal swith port. Nowadays, all switches have a separate port on which all network traffic in the switch can be monitored. (Or at least a normal port can be configured to pass on ALL traffic in the switch.) That way it is possible to detect what packets are going where. To monitor if a packet is making it through (or not making it through), a tcpdump should be run both in front and behind the firewall. Some examples of tcpdump in action (once again the documentation of tcpdump gives all the options):

`tcpdump`	will dump a summary of every packet on the network
`tcpdump host dns1`	will dump all traffic to and from dns1
`tcpdump src host dns1`	will summarize all traffic from dns1
`tcpdump dst host dns1`	will summarize all traffic going to dns1
`tcpdump host admin1 and host dns1`	will summarize all traffic between admin1 and dns1

Typical tcpdump output:

```
21:39:36.697883 192.168.0.1.2123 > 192.168.0.2.12345: S 394849967:394849967(0)
win 5840 <mss 1460,sackOK,timestamp 52724[|tcp]> (DF)
21:39:36.697883 192.168.0.2.12345 > 192.168.0.1.2123: S 400016818:400016818(0)
ack 394849968 win 5792 <mss 1460,sackOK,timestamp 51928[|tcp]> (DF)
21:39:36.697883  192.168.0.1.2123  >  192.168.0.2.12345:  .  ack  1  win  5840
<nop,nop,timestamp 52724 51928> (DF)
```

What this shows us is three captured packets. A SYN a SYN/ACK and finally an ACK. In short some details; the first field is the timestamp, the second the source IP and port, then the destination IP and port. Following fields give SYN, ACK, Don't Fragment, Sequence numbers, window sizes and a lot of other information which is very well described in the manual.

Finally, the last tool I want to present is netcat (nc) that can be used to initiate and receive connections (UDP/TCP). The examples should be self-explanatory:

`nc dns1 53`	connect to TCP port 53 on dns1
`nc -u dns1 53`	connect to UDP port 53 on dns1
`nc -l 24`	listen on TCP port 24 and accept connections

`nmap`, `netcat` and `tcpdump` are very useful tools to find out which packets make it where and what ports are open and reachable.

A word on using groups instead of address ranges. The strength of address ranges is that they leave room for future expansion. If more machines are added of the same type, the addresses are available in the address range. Yet this could open up security gaps in the firewall since it allows packets to come in for machines that don't exist, if filtering is done on address ranges. A group of workstations specifically requires each new server to be added to the firewall rulebase. Thus, administrators will have to actively think about opening an access path through the firewall. If address ranges are used in the firewall for allowing packets to pass through and the reserved IP addresses are used for other servers than originally intended, those machines could be vulnerable on that original service. New machines need to be added to the Firewall 1 group.

c. Objects to be constructed.

Individual Networking Objects:

OBJECT	NAME	X.Y.Z._
Border Router	brouter	.1
Main Firewall	mainfw	.2 .33 .65 .129
SMTP/POP3 mail	Mail server	.34
Primary DNS server	dns1	.35
Secondary DNS server	dns2	.36
Web server	web1	.70

Web server	web2	.71
Web server	web3	.72
Administrative console	admin1	.130
Administrative console	admin2	.131
Administrative console	admin3	.132
Backup	Backup server	.135
Logging server	logger	.136
Database server	database	.140
Internal Firewall	intfw	.150 .151 .180
Corporate DB clearance	corp1DB	.152
Corporate DB clearance	corp2DB	.153
Corporate DB clearance	corp3DB	.154
Corp no DB clearance	corp4	.155
Corp no DB clearance	corp5	.156
Corp no DB clearance	corp6	.157
Corp no DB clearance	corp7	.158
Corp no DB clearance	corp8	.159
VPN channel host	vpn1	.181
business partner DB cleared host	partner1	routed through VPN
business partner DB cleared host	partner2	routed through VPN
business partner IPSEC	vpnpartn	
banking computer SET	bank	

Groups:

OBJECT	NAME	HOSTS
Webserver pool	webpool	web1, web2, web3
DMZ	dmz	mail, dns1, dns2
Administrative	admin	admin1, admin2, admin3, backup, logger
Corporate DB clearance	corpDB	corp1DB, corp2DB, corp3DB
Corporate all	corpall	corp1DB - corp8
Partner DB clearance	partDB	partner1, partner2

Logical Servers:

OBJECT	GROUP	NAME	IP ADDRESS
Webserver	webpool	www	X.Y.Z.66

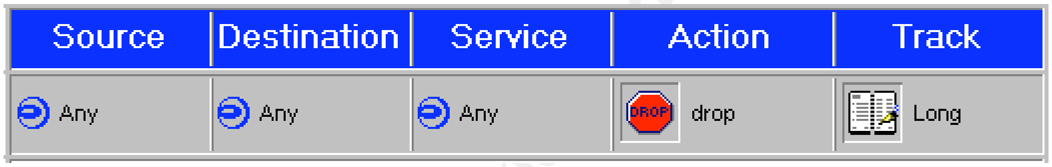
Address Range:

OBJECT	NAME	RANGE
Our local address range	local	X.Y.Z.0 - X.Y.Z.255
Business partners	partnrng	X.Y.Z.200 - X.Y.Z.250
Corporate machines	corporng	X.Y.Z.152 - X.Y.Z.180
Internal machines	internal	X.Y.Z.128 - X.Y.Z.255

d. Creating the rulebase.

d.1

Starting off with the first specific rule of the security policy, implementing a default deny rule at the end of the main Firewall rulebase:



Source	Destination	Service	Action	Track
Any	Any	Any	drop	Long

Testing this can be done by trying to send arbitrary packets through the firewall and testing if they are show up at the other side. The logs should indicate that this packet has been dropped. From the DMZ make a connection attempt (with netcat) to a web server and the database server:

```
nc web1 80
nc database 25
```

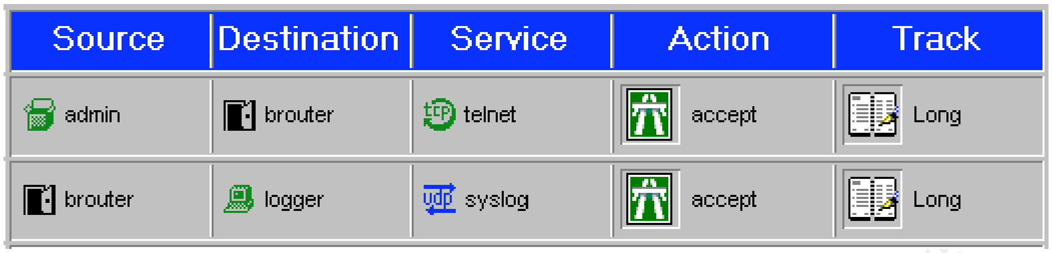
Using tcpdump on the web server segment and the internal segment should come up blank (since the packets will be dropped by the firewall):

```
tcpdump host web1
tcpdump host database
```

The log files of the firewall will show that the packets have been dropped.

d.2

Next are the connections to and from the border router. The administrator machines are allowed to telnet to the border router for maintenance and the border router is allowed to log to the logserver through syslog:



Source	Destination	Service	Action	Track
admin	brouter	telnet	accept	Long
brouter	logger	syslog	accept	Long

Obviously these two rules are placed above the default deny rule, since that rule matches everything and thus drops everything.

Testing the two new rules with a telnet to the border router from the administrative network:

```
        nc brouter 23
```

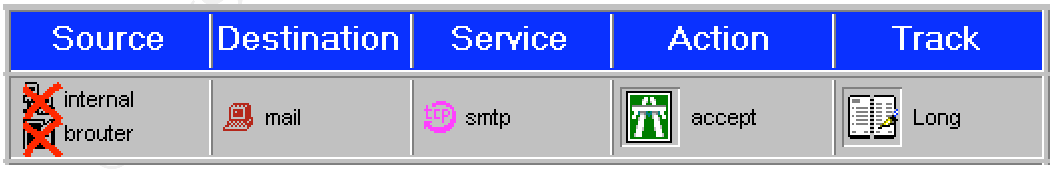
This should come up with a login prompt on the border router. The logs will show the packet has been accepted. To be very sure that all other connections get blocked, connecting from different machines or connecting to different ports could be tried:

```
        nc brouter 23   from the web server pool
        nc brouter 24   from the admin network
```

Both examples should be dropped and logged by the firewall. This is because of the default deny rule. The default deny rule will always be the last rule in the rulebase. Implementing and testing syslog from the border router is described later in this document.

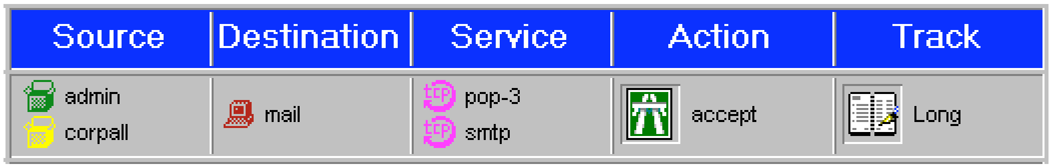
d.3

Connections to and from the Mail server. In the security policy this is described as the Internet, the webserverpool, the administrative network and the corporate network. This firewall cannot check connections inside the DMZ, this is because those machines are on the same segment. In general, internal machines are not allowed to connect, except for the corporate and the administrative network. This could be described as everyone may connect, except internal (and the border router):



Source	Destination	Service	Action	Track
✗ internal ✗ brouter	mail	smtp	accept	Long

Yet, this rule will prevent both administrative and corporate to send email. So above this rule there should be a rule to specifically allow those groups to connect. Furthermore, when reading a little further corporate and administrative are the only ones allowed to make POP3 connections to the mail server. This can be included in this rule too:



Source	Destination	Service	Action	Track
admin corpall	mail	pop-3 smtp	accept	Long

Once again, this last rule should be above the previous one. Implicitly, this allows the main firewall to send email to administrators if necessary.

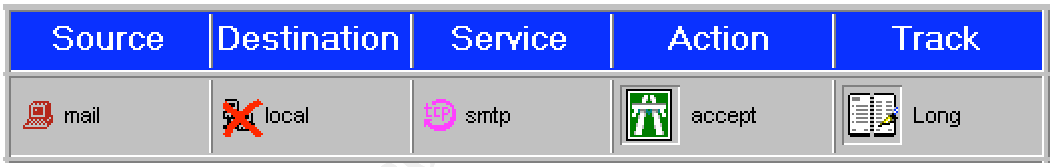
Testing the two new rules is done by trying to connect to the mail server from the corporate, the administrative network, the webserver pool or the internet (nc mail 25). The connection should be fine. Trying to connect from the database should fail, this, again, can be viewed in the firewall logs. Testing pop-3 can be done as follows from the administrative and corporate network:

```
nc mail 110
```

Other connections should all be refused, e.g.:

```
nc mail 123
```

Furthermore, the mail server is allowed to make SMTP connections to all non-local machines. This is for delivering mail:



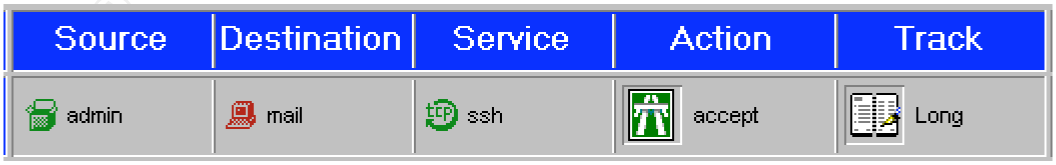
Source	Destination	Service	Action	Track
mail	local	smtp	accept	Long

To test this rule we use netcat:

```
nc <non-local IP> 25
```

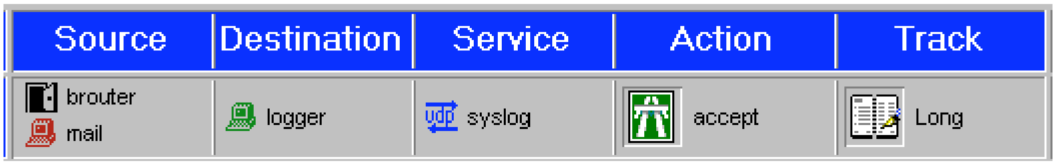
This connection should succeed and be logged by both the main firewall and the syslog server.

Two more rules for the mail server. System administration is allowed to make SSH connections to the mail server for administrative and backup purposes:



Source	Destination	Service	Action	Track
admin	mail	ssh	accept	Long

Also, the mail server is allowed to make connections to the syslog server. mail can be added to the syslog rule of the border router:

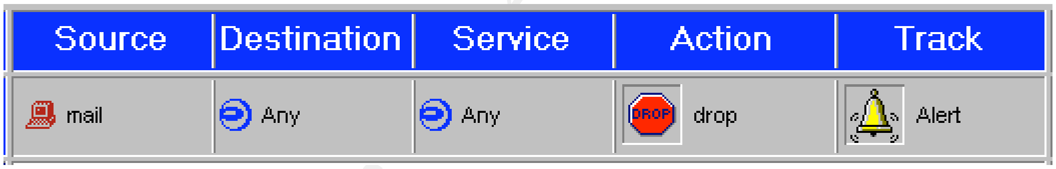


Source	Destination	Service	Action	Track
brouter mail	logger	syslog	accept	Long

It is no use to create a rule that allows the mail server to make DNS requests. This is because the mail server and the DNS servers are on the same physical network segment. DNS requests never reach the firewall from the mail server.

Testing the two new rules is done by making a ssh connection from the administrative machines to the mail server (ssh mail). This should give a login prompt. Connections on other ports should be refused and show up in the firewall logs (egg. nc mail 321). Generating syslog messages can easily be done by sending an email. The mail server should log this to the log server. If this fails, a tcpdump can be run on the DMZ and on the internal network to see if the UDP packet is on the network: `tcpdump host mail`. If it's on the DMZ but not on the internal network the firewall logs should give the clue.

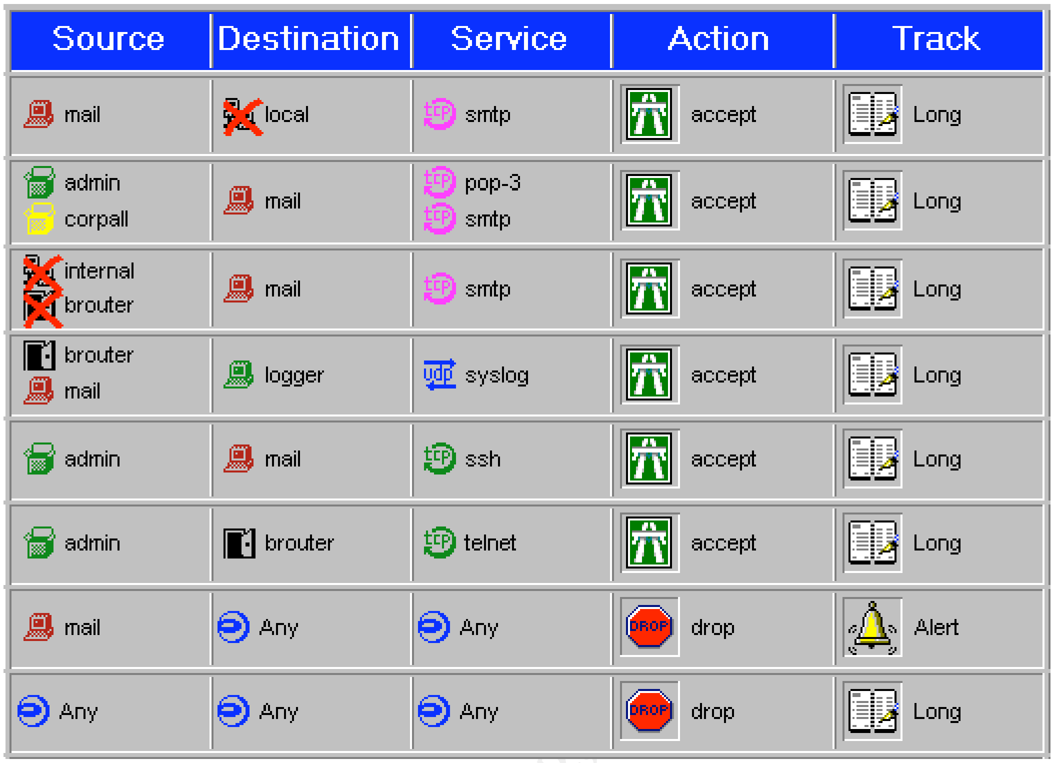
Finally, because the mail server is not allowed to make any other connections except DNS and syslog, it should be considered strange if a different connection attempt is detected. This could be a sign of an intrusion of in the DMZ. To alert the system administrators of this, a rule should be created which warns as soon as a connection attempt from the mail server is detected. This rule should be at the bottom, just before the default deny rule:



Source	Destination	Service	Action	Track
mail	Any	Any	drop	Alert

To test: just try to connect to the database from the mailserver. This should create a firewall log entry and send an email to the administrator.

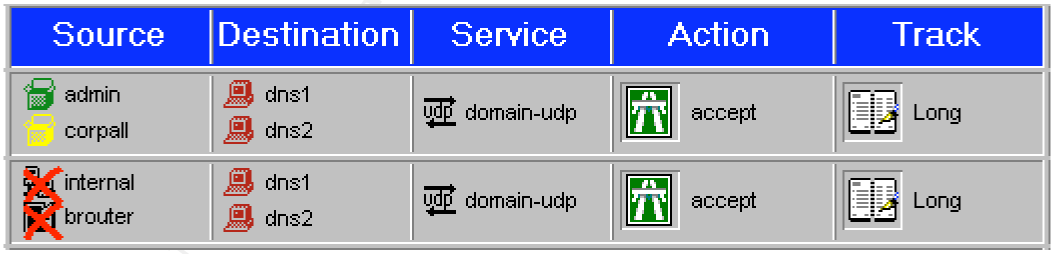
Current rulebase:



Source	Destination	Service	Action	Track
mail	local	smtp	accept	Long
admin corpall	mail	pop-3 smtp	accept	Long
internal brouter	mail	smtp	accept	Long
brouter mail	logger	syslog	accept	Long
admin	mail	ssh	accept	Long
admin	brouter	telnet	accept	Long
mail	Any	Any	drop	Alert
Any	Any	Any	drop	Long

d.4

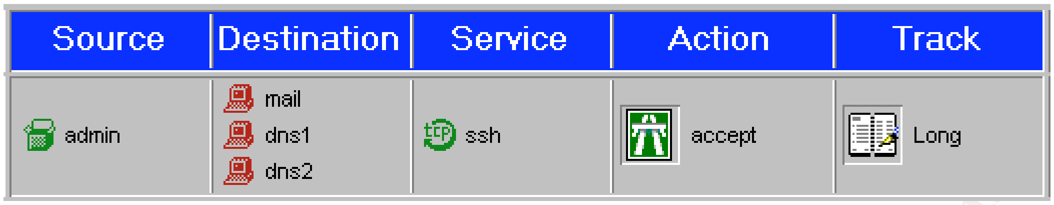
Next are the DNS servers. The DNS servers accept DNS lookup (UDP) requests from the same group as the mail server accepts it's SMTP requests from. The same story goes here as it did with the mail server. Rules:



Source	Destination	Service	Action	Track
admin corpall	dns1 dns2	domain-udp	accept	Long
internal brouter	dns1 dns2	domain-udp	accept	Long

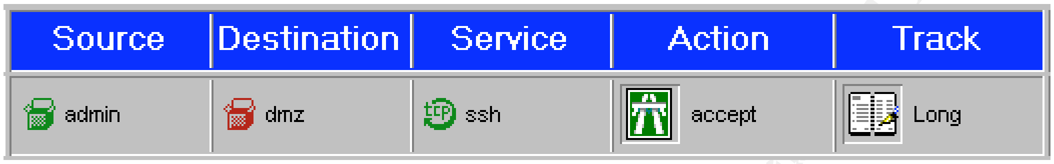
Tests can also be conducted in the same manner: making DNS requests from various locations and checking the logs for results. Also, denies should be tested from various locations for various port numbers.

SSH connections from the administrative group can be added to the administrative rule for the mail server:



Source	Destination	Service	Action	Track
admin	mail dns1 dns2	ssh	accept	Long

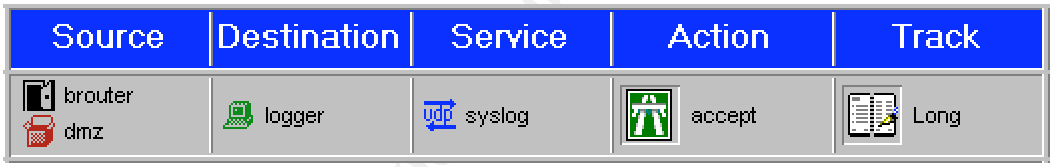
Which can be substituted by:



Source	Destination	Service	Action	Track
admin	dmz	ssh	accept	Long

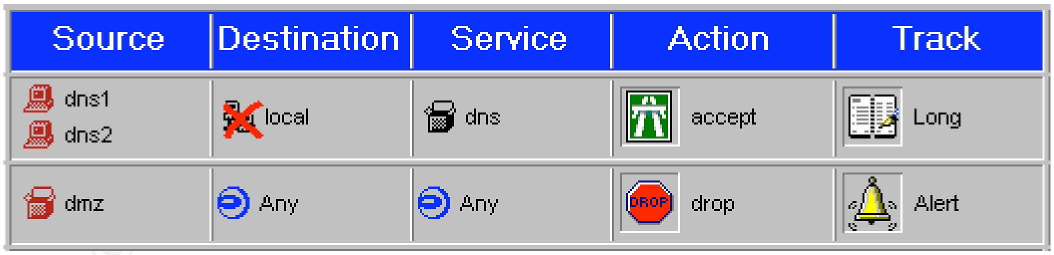
Testing is once again done in the same way as with the mail server. Initiate SSH connections from the administrative machines to both the DNS servers. Connecting to a different port should fail and produce a log entry in the firewall logs. Connection attempts from anywhere but the administrative machines should also fail.

Similar modification could be applied to logging. The DNS servers are allowed to log to the logger, which makes up for the whole DMZ to log through syslog:



Source	Destination	Service	Action	Track
brouter dmz	logger	syslog	accept	Long

Next, DNS servers are allowed to make all DNS (TCP/UDP) connections to everyone except the local network (our class C). If the DNS servers try to initiate any other connection a warning should be given, again analogous to the mail server:

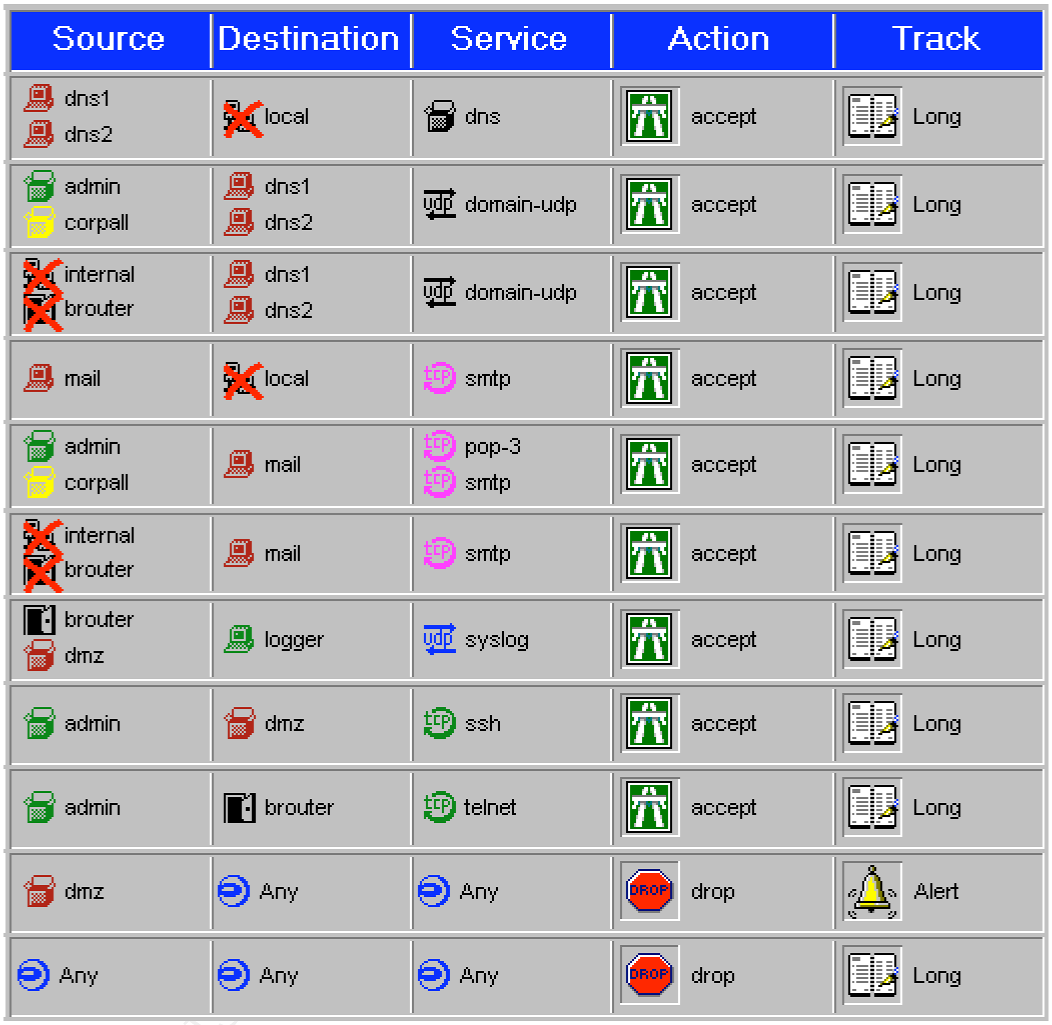


Source	Destination	Service	Action	Track
dns1 dns2	local	dns	accept	Long
dmz	Any	Any	drop	Alert

Testing the connections made by the DNS servers is done as follows. Generating syslog information on the DNS servers should result in successful logging on the logserver. This can be tested and debugged in the way described for the mail server. Furthermore, DNS (UDP/TCP) connections from the DNS servers should be successful for the whole Internet except for the local addresses. Any other connection initiated by the DMZ should sound an alarm to the administrators.

To allow for zone transfers between the primary and the secondary DNS, nothing needs to be done. This is because the servers are both on the same network segment. The same with DNS requests from the mail server.

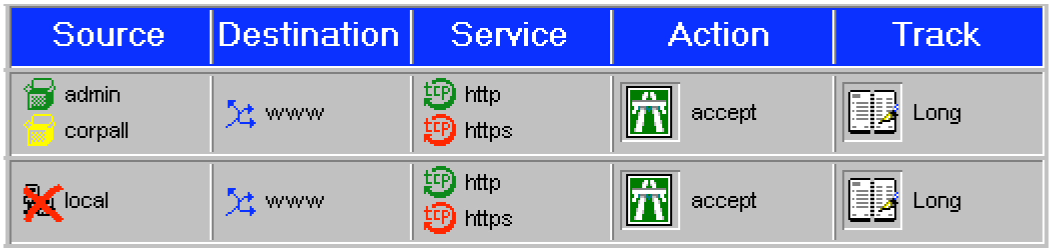
Current rulebase:



Source	Destination	Service	Action	Track
dns1 dns2	local	dns	accept	Long
admin corpall	dns1 dns2	domain-udp	accept	Long
internal brouter	dns1 dns2	domain-udp	accept	Long
mail	local	smtp	accept	Long
admin corpall	mail	pop-3 smtp	accept	Long
internal brouter	mail	smtp	accept	Long
brouter dmz	logger	syslog	accept	Long
admin	dmz	ssh	accept	Long
admin	brouter	telnet	accept	Long
dmz	Any	Any	drop	Alert
Any	Any	Any	drop	Long

d.5

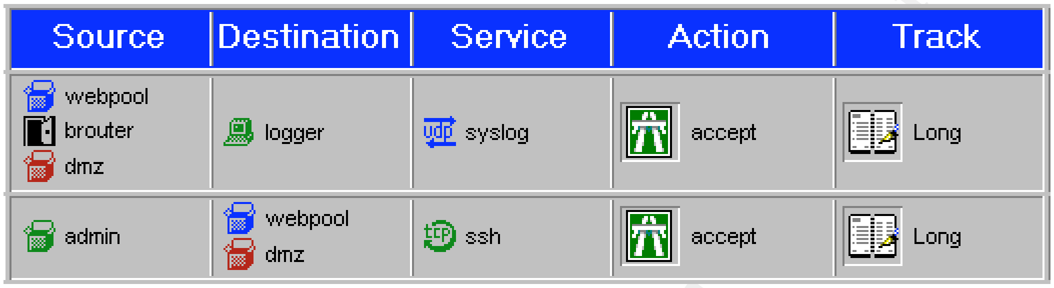
Connections to and from the webserver pool. HTTP and HTTPS connections are accepted from the Internet, administrative network and the corporate network. The Internet is everything NOT local:



Source	Destination	Service	Action	Track
admin corpall	www	http https	accept	Long
local	www	http https	accept	Long

This can be tested in by connecting with HTTP and HTTPS to www from the Internet, the admin and the corpall groups. Connections should all be accepted. Connecting to forbidden ports or from a non-authorized host (e.g. database) should result in a drop and a log entry.

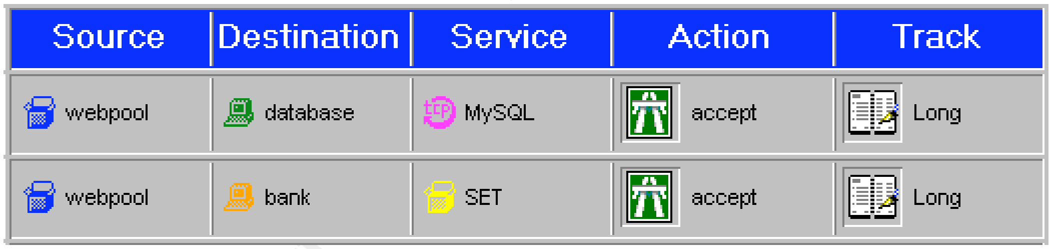
Next, SSH and syslog are added in the same way as described for the DMZ servers above (webpool added to existing rules):



Source	Destination	Service	Action	Track
webpool brouter dmz	logger	syslog	accept	Long
admin	webpool dmz	ssh	accept	Long

These rules are tested in the same way as described above. Generate logs in the webpool and verify that they show up in the syslog server. If this fails, firewall logs and tcpdump output may give a clue as to where the packets go. Connect ssh to all the webservers to verify that the connection is allowed. Also test from the internet and, for example corporate, if SSH connections are refused.

Next, MySQL connections to the database and SET connections to the bank should be permitted:

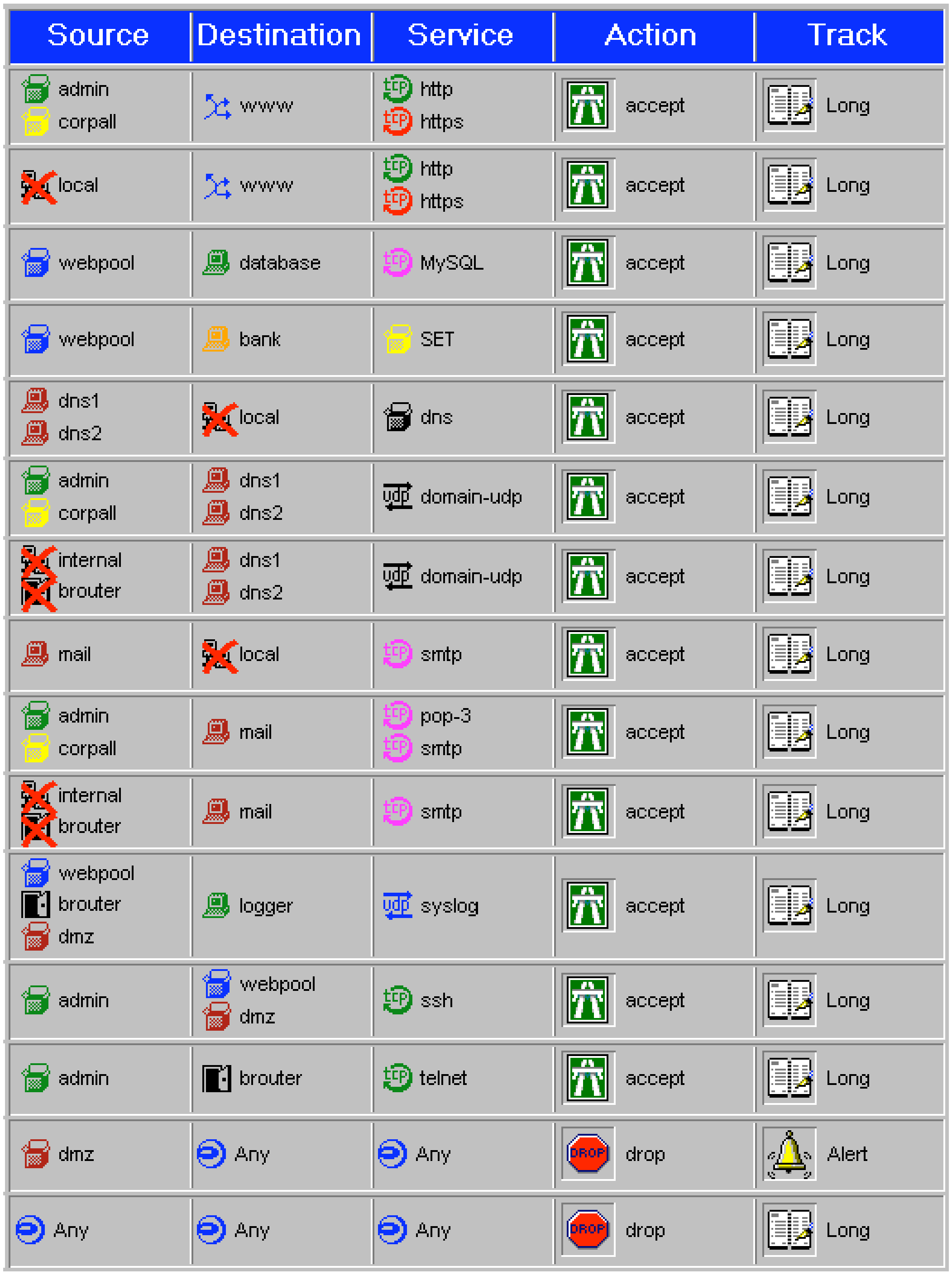


Source	Destination	Service	Action	Track
webpool	database	MySQL	accept	Long
webpool	bank	SET	accept	Long

Tests are performed by making MySQL requests to the database from the webservers. Attempts to make different connections to the database from the webserver pool should fail. Further testing on the database will be performed later on. Connection attempts from the webserver pool to the bank should work fine. Connection logs provide proof.

Finally, DNS requests and SMTP connections are already allowed by the rulebase.
Testing this can be done using, e.g. nslookup from the webservers to resolve an arbitrary hostname. Sending mail should work as a test for SMTP. If these tests fail, firewall logs should provide a clue as to what is happening. Once again tcpdump can be used on both the webserver segment and the DMZ segment to check what traffic is

Current rulebase:



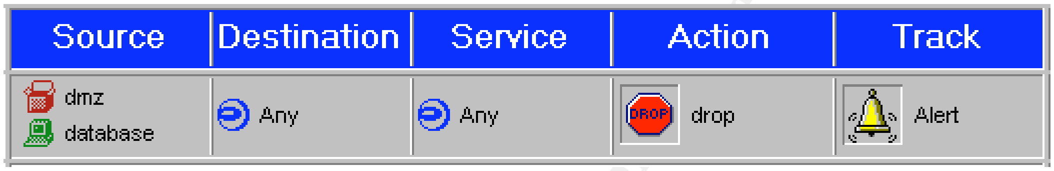
Source	Destination	Service	Action	Track
admin corpall	www	http https	accept	Long
local	www	http https	accept	Long
webpool	database	MySQL	accept	Long
webpool	bank	SET	accept	Long
dns1 dns2	local	dns	accept	Long
admin corpall	dns1 dns2	domain-udp	accept	Long
internal brouter	dns1 dns2	domain-udp	accept	Long
mail	local	smtp	accept	Long
admin corpall	mail	pop-3 smtp	accept	Long
internal brouter	mail	smtp	accept	Long
webpool brouter dmz	logger	syslog	accept	Long
admin	webpool dmz	ssh	accept	Long
admin	brouter	telnet	accept	Long
dmz	Any	Any	drop	Alert
Any	Any	Any	drop	Long

d.6

The customer database server already accepts MySQL connections from the webpool. Yet there are certain corporate machines and certain business partner machines which are allowed to use MySQL queries on the database. Those machines are in groups: corpDB and partDB. (Note that machines of business partners, coming in through the VPN tunnel have non-local IP addresses and are thus not routed through the main firewall. They can only enter through the VPN host.) This, however should be handled by the internal firewall, since both connections are not going through the main firewall. Still, tests should be done to ensure no one from the outside can connect to the customer database. This can be done by trying to make any kind of connection to the database from the Internet. This should all fail and be logged by the firewall.

Once again, SSH connections from the administrative machines and syslog connections to the log server are allowed. Yet, since the database is on the same physical segment as the administrative machines, no rule has to be added for the main firewall. The packets will never reach the main firewall. This can be tested by simply making the connections. If the connections fail, it isn't because of the firewall and tcpdump should be used to check what's going on.

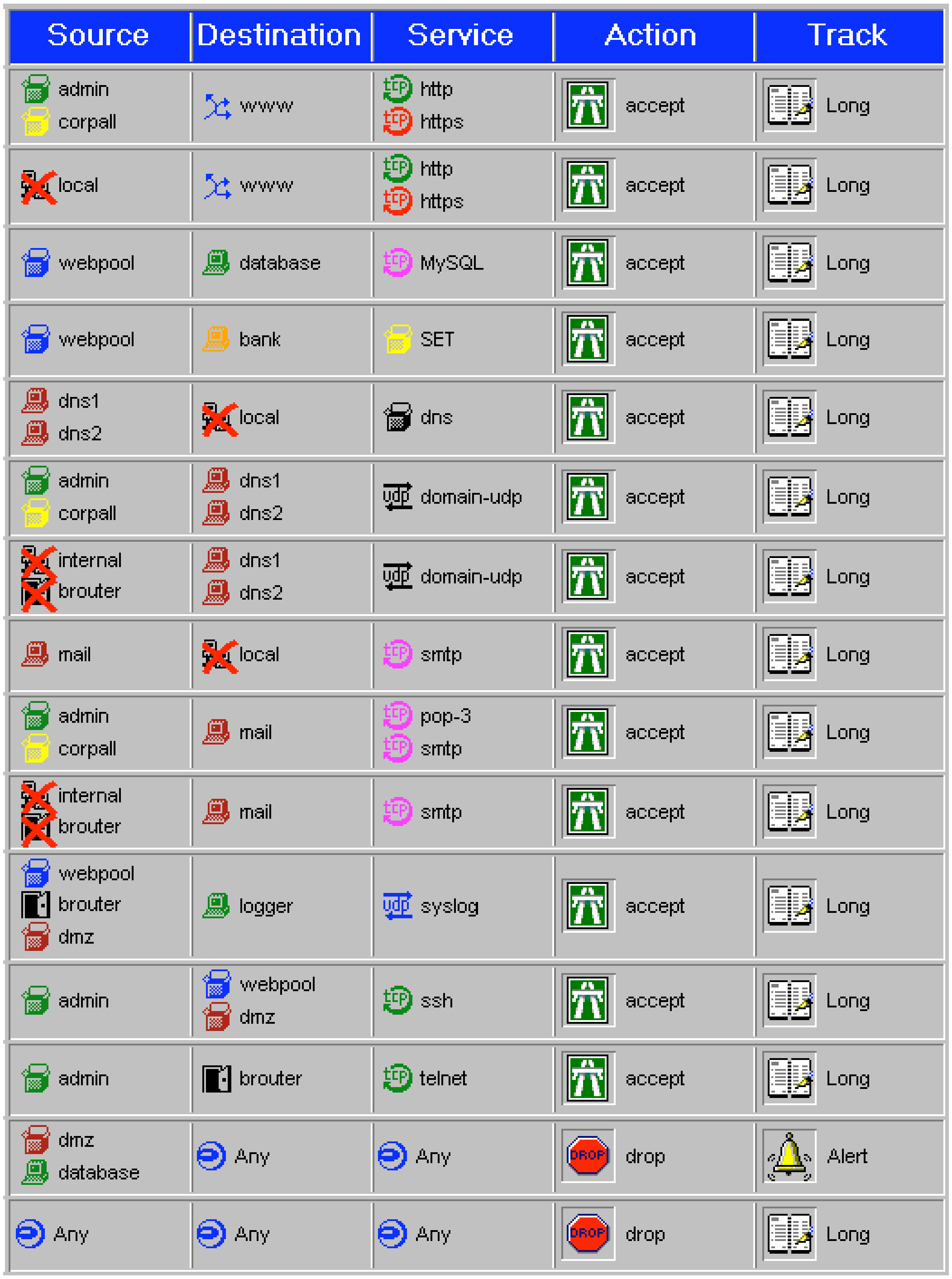
Any other connection attempts by the database server will trigger an intrusion warning, equal to the DMZ. Rule is added as follows:



Source	Destination	Service	Action	Track
dmz database	Any	Any	drop	Alert

Test this by trying to telnet to a machine on the internet. An email should arrive with the administrator.

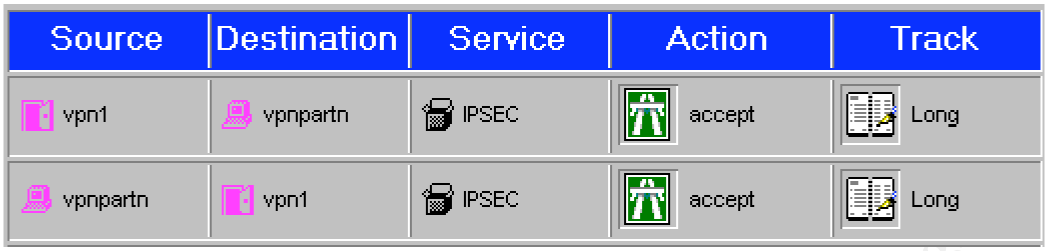
Current rulebase:



Source	Destination	Service	Action	Track
admin corpall	www	http https	accept	Long
local	www	http https	accept	Long
webpool	database	MySQL	accept	Long
webpool	bank	SET	accept	Long
dns1 dns2	local	dns	accept	Long
admin corpall	dns1 dns2	domain-udp	accept	Long
internal brouter	dns1 dns2	domain-udp	accept	Long
mail	local	smtp	accept	Long
admin corpall	mail	pop-3 smtp	accept	Long
internal brouter	mail	smtp	accept	Long
webpool brouter dmz	logger	syslog	accept	Long
admin	webpool dmz	ssh	accept	Long
admin	brouter	telnet	accept	Long
dmz database	Any	Any	drop	Alert
Any	Any	Any	drop	Long

d.7

The VPN encrypter/decrypter host hast to be able to accept and initiate IPSEC connections with the VPN encrypter/decrypter host of the business partner on the Internet. IPSEC is a combination of the AH, the ESP and the SKIP protocol and used in most VPN solutions nowadays. Checkpoints Firewall 1 has IPSEC as a protocol group for VPN communication:



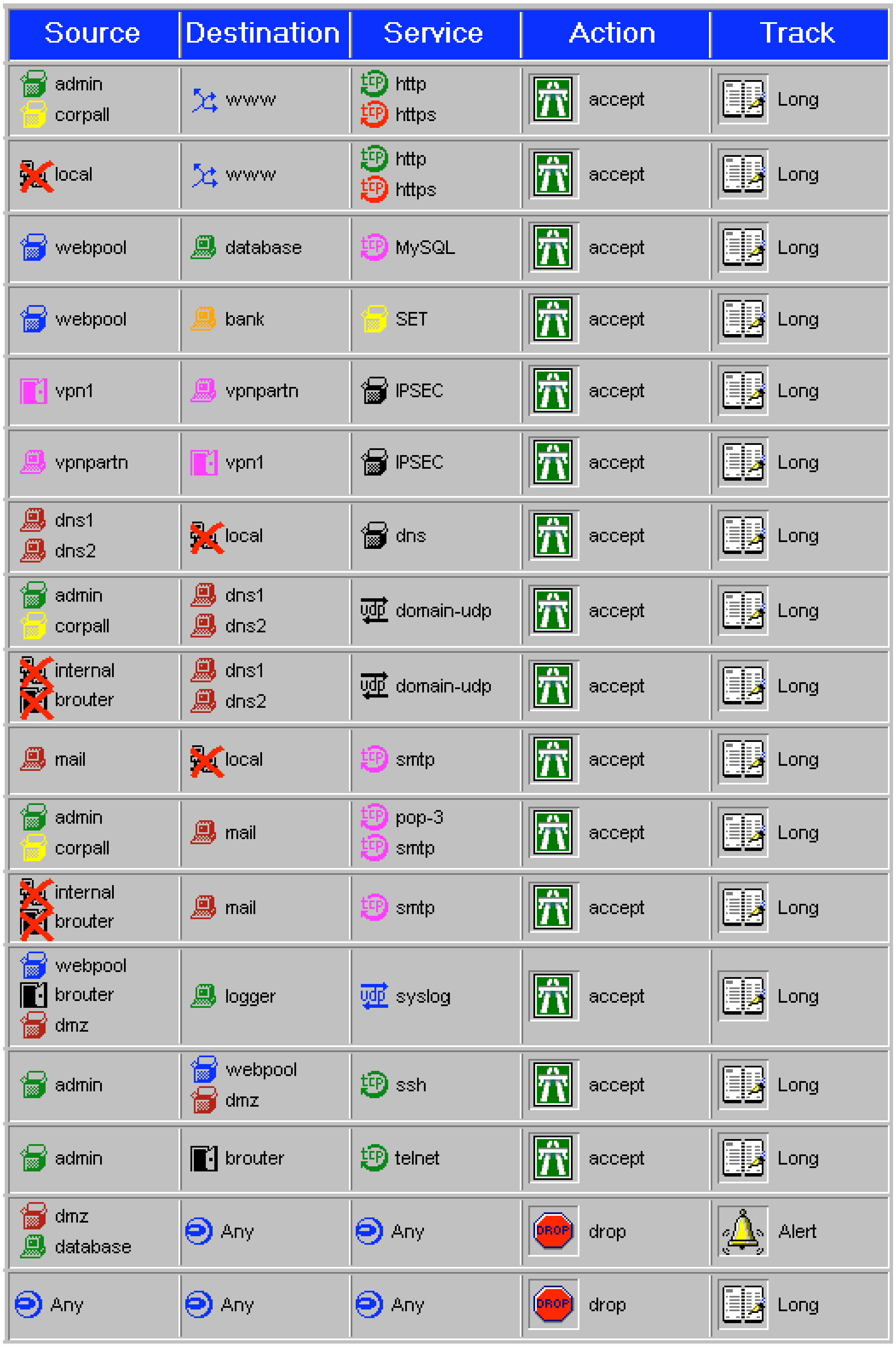
Source	Destination	Service	Action	Track
vpn1	vpnpartn	IPSEC	accept	Long
vpnpartn	vpn1	IPSEC	accept	Long

This connection has to go through the main firewall to get in! This can be tested by initiating and accepting the vpn connection. Also, it should be tested if any other connections get through the main firewall. This can be done by initiating connections to the vpn1 host from the internet (e.g. `nc vpn1 22`, from outside).

Again, also the vpn1 host should be able to syslog to the log server and should be able to accept SSH connections from the administrative network. Yet both these connections don't go through the main firewall. They get handled by the internal firewall.

Since the partDB hosts have access to the customer database we want to ensure they can only enter the network through the VPN tunnel and not through the Internet. Yet, since the partDB group enters through the VPN tunnel, no rule was created in the main firewall to allow them access to the database. So whenever a packet from the Internet, with a source address of the partDB group, comes in it is already denied (since there was no specific rule allowing them in.)

Current rulebase:

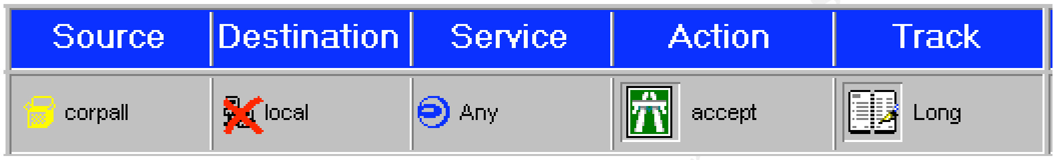


Source	Destination	Service	Action	Track
admin corpall	www	http https	accept	Long
✗ local	www	http https	accept	Long
webpool	database	MySQL	accept	Long
webpool	bank	SET	accept	Long
vpn1	vpnpartn	IPSEC	accept	Long
vpnpartn	vpn1	IPSEC	accept	Long
dns1 dns2	✗ local	dns	accept	Long
admin corpall	dns1 dns2	domain-udp	accept	Long
✗ internal ✗ brouter	dns1 dns2	domain-udp	accept	Long
mail	✗ local	smtp	accept	Long
admin corpall	mail	pop-3 smtp	accept	Long
✗ internal ✗ brouter	mail	smtp	accept	Long
webpool brouter dmz	logger	syslog	accept	Long
admin	webpool dmz	ssh	accept	Long
admin	brouter	telnet	accept	Long
dmz database	Any	Any	drop	Alert
Any	Any	Any	drop	Long

d.8

For the corporate network most connections are filtered by the internal firewall. Connections to the corporate network are handled by the internal firewall and so is syslog access. Database access and access to the business partners machines is also handled by the internal firewall. Webpool access (HTTP/HTTPS), mail (SMTP) and DNS access are already specified in the rulebase of the main firewall. These can all be tested in the standard way. Trying to connect to each of those services should succeed. Other connections to these machines should fail. Firewall logs show the specifics.

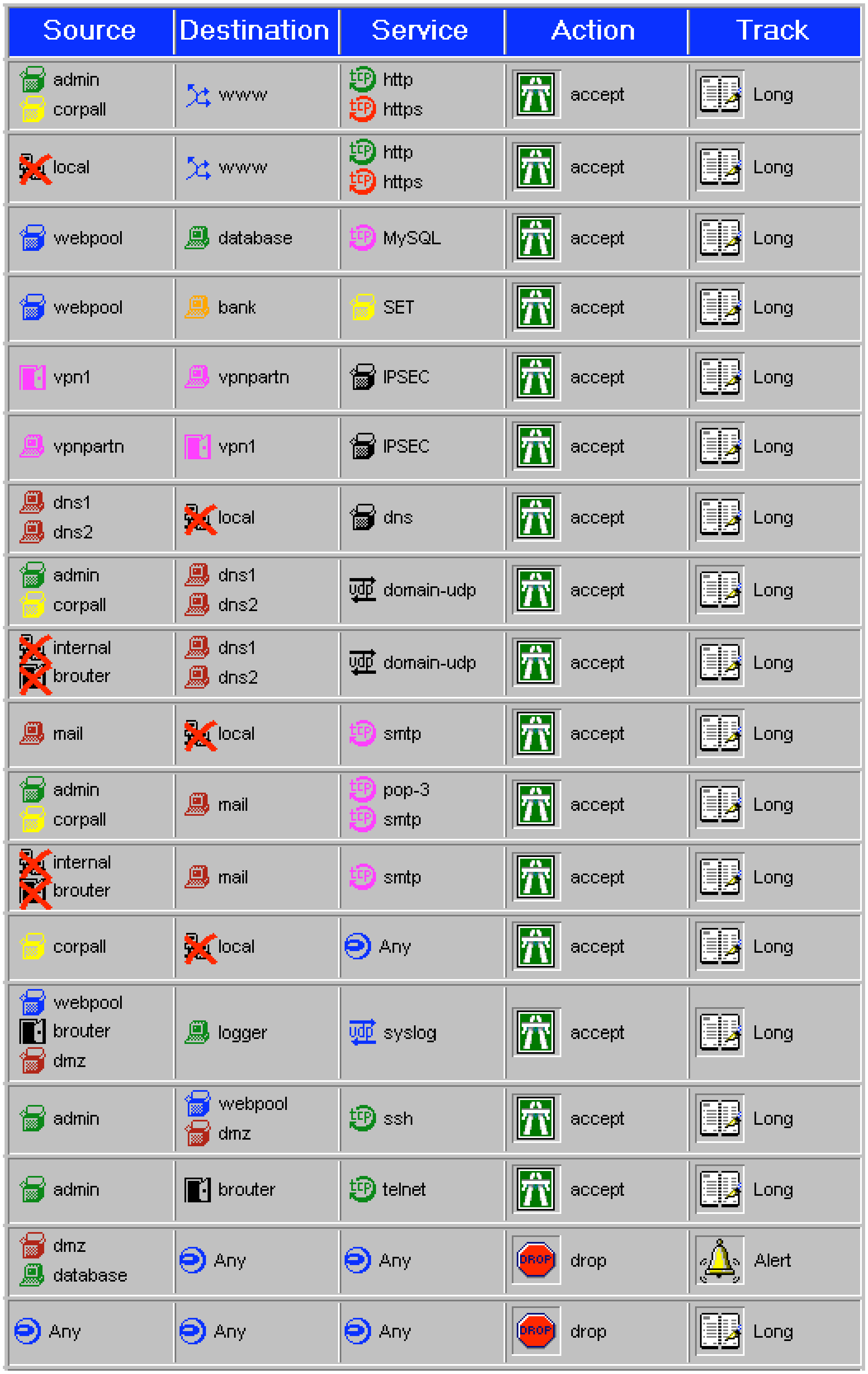
To allow for corporate machines to connect to the internet, a rule needs to be created to account for all non-local machines:



Source	Destination	Service	Action	Track
corpall	local	Any	accept	Long

Testing should be done if connections to the internet can be made, but also if web, mail and dns are still fully working. (This shouldn't be a problem though since this rule doesn't match any of those three services, and thus doesn't get applied.)

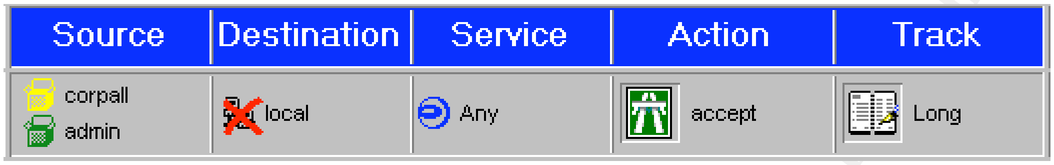
Current rulebase:



Source	Destination	Service	Action	Track
admin corpall	www	http https	accept	Long
local	www	http https	accept	Long
webpool	database	MySQL	accept	Long
webpool	bank	SET	accept	Long
vpn1	vpnpartn	IPSEC	accept	Long
vpnpartn	vpn1	IPSEC	accept	Long
dns1 dns2	local	dns	accept	Long
admin corpall	dns1 dns2	domain-udp	accept	Long
internal brouter	dns1 dns2	domain-udp	accept	Long
mail	local	smtp	accept	Long
admin corpall	mail	pop-3 smtp	accept	Long
internal brouter	mail	smtp	accept	Long
corpall	local	Any	accept	Long
webpool brouter dmz	logger	syslog	accept	Long
admin	webpool dmz	ssh	accept	Long
admin	brouter	telnet	accept	Long
dmz database	Any	Any	drop	Alert
Any	Any	Any	drop	Long

d.9

To allow for the administrative network to connect to the internet the admin group is added to the rule created for the corporate network:



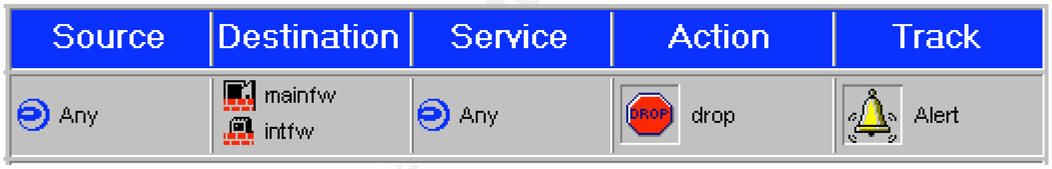
Source	Destination	Service	Action	Track
corpall admin	local	Any	accept	Long

Again, it should be tested that the administrative machines can connect to the internet.

Rules for www, mail (SMTP and POP-3) and dns have already been created and tested above. All SSH administrative connections have been accounted for already (except for the ones handles by the internal firewall). Logging, backup and database is on the same network segment and doesn't require any rules.

d.10

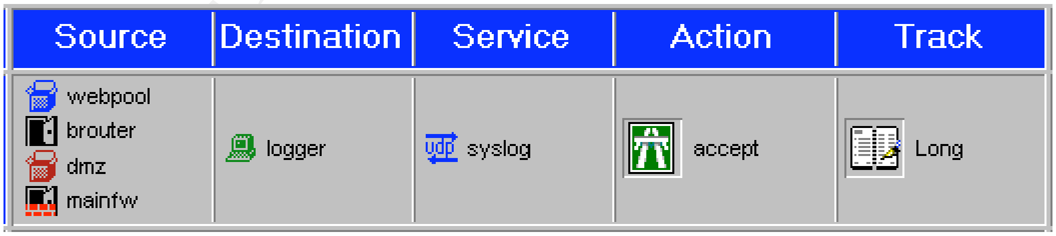
Both firewall don't accept any connections from anywhere. If someone is trying to initiate a connection, that should sound a warning:



Source	Destination	Service	Action	Track
Any	mainfw intfw	Any	drop	Alert

This rule is placed at the end, just before the default deny rule. Test this by, for example, telnet to the main firewall. A warning should be generated in the form of an email to the administrator.

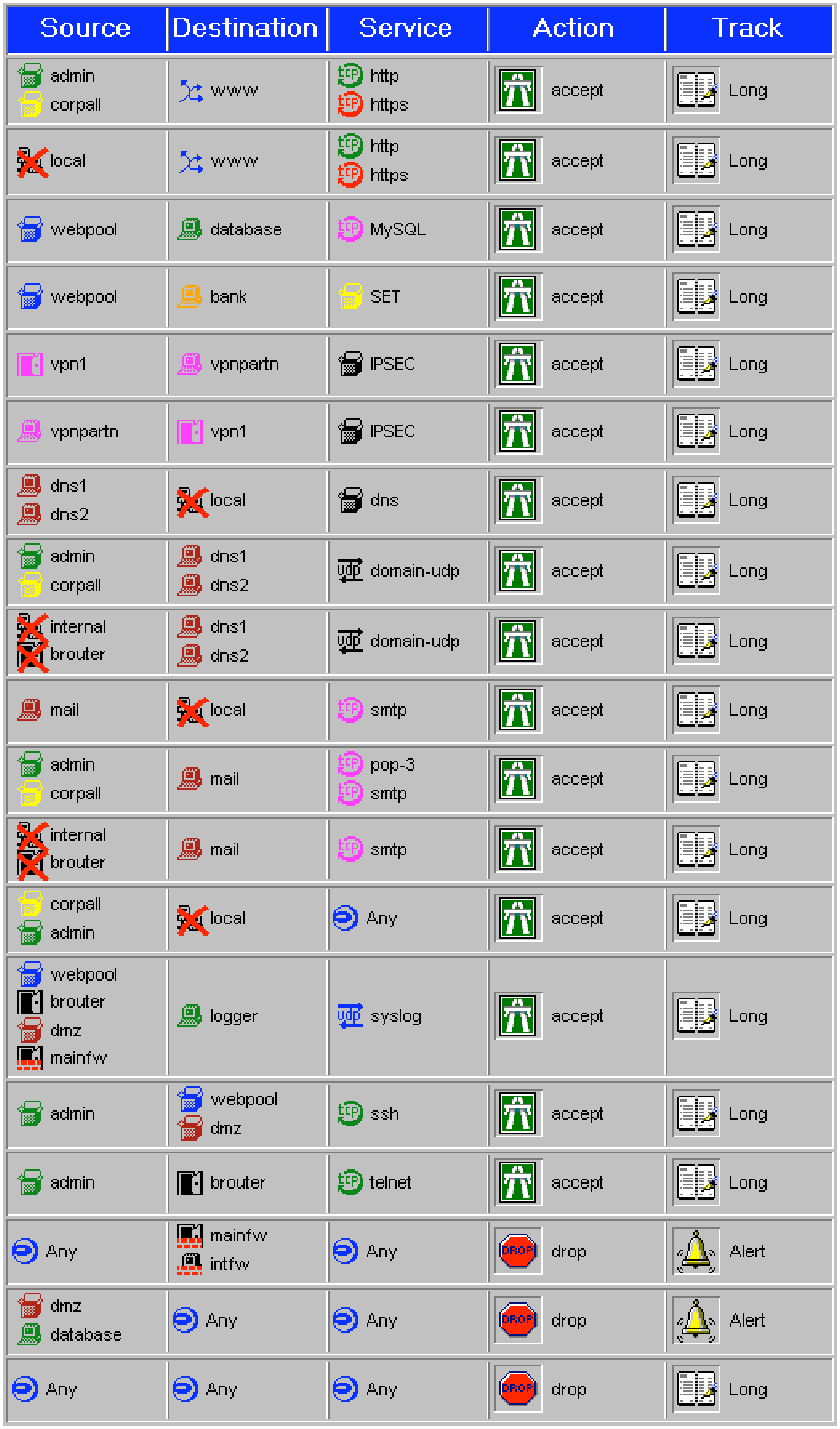
An entry is added for the main firewall to make syslog connections:



Source	Destination	Service	Action	Track
webpool brouter dmz mainfw	logger	syslog	accept	Long

Once again this is tested by having the firewall generate logging information and checking if it shows up on the log host.

Final rulebase:



Source	Destination	Service	Action	Track
admin corpall	www	http https	accept	Long
local	www	http https	accept	Long
webpool	database	MySQL	accept	Long
webpool	bank	SET	accept	Long
vpn1	vpnpartn	IPSEC	accept	Long
vpnpartn	vpn1	IPSEC	accept	Long
dns1 dns2	local	dns	accept	Long
admin corpall	dns1 dns2	domain-udp	accept	Long
internal brouter	dns1 dns2	domain-udp	accept	Long
mail	local	smtp	accept	Long
admin corpall	mail	pop-3 smtp	accept	Long
internal brouter	mail	smtp	accept	Long
corpall admin	local	Any	accept	Long
webpool brouter dmz mainfw	logger	syslog	accept	Long
admin	webpool dmz	ssh	accept	Long
admin	brouter	telnet	accept	Long
Any	mainfw intfw	Any	drop	Alert
dmz database	Any	Any	drop	Alert
Any	Any	Any	drop	Long

Above the final rulebase for the main firewall is presented. Although the firewall has an implicit 'default deny' rule, the rule is added here explicitly to allow for long logging. The rules are created to only allow valid connections to be made. This does not mean that all attacks get blocked. If, for example, the web servers would be attacked by a double-dot attack (one or more "../" in the URL to retrieve a file not meant to be exported. This way a path to a higher directory on the web server can be specified.) If the double-dot request gets made through a valid HTTP request, the firewall won't block this. Only by keeping security patches up to date, these 'application level' attacks can be prevented. Just a firewall doesn't guarantee a safe network!

4. Border Router

Since the focus was on the main firewall, this part will be short but complete. This part will only describe which filtering rules will be installed on the router.

The border router is where we will take care of the first rule of the security policy:

1. block company's address spoofed packets going inbound
2. block private address and network 127 packets going inbound
3. block source routed packets going inbound
4. block packets with non-local addresses going outbound (spoofed from inside)

Cisco IOS provides both standard ACL's (filtering based on source IP only) and extended ACL's (filtering on source IP, source port, destination IP, etc.). The inbound filters can all be applied as a standard ACL ingress filter on the outside interface (so packets get dropped before they get routed). A Cisco standard ACL syntax block:

```
IP access-list standard/extended [name]
    permit/deny {test condition}
!comment is exclamation mark
```

The first two filtering rules can be applied to the ingress (inbound) filter, everything will be logged:

```
IP access-list standard INBOUND
    ! private address space and 127 addresses
    deny  10.0.0.0    0.255.255.255  log
    deny  172.16.0.0  0.15.255.255   log
    deny  192.168.0.0 0.0.255.255    log
    deny  127.0.0.0   0.255.255.255  log
    deny  224.0.0.0   7.255.255.255  log
    ! our own address space
    deny  X.Y.Z.0     0.0.0.255      log
    ! rest is ok
    permit any
```

The third filtering rule is a configuration option and applied as follows:

```
brouter# config t
brouter(config)# no ip source-route
```

This can be applied to the external interface of the Border Router:

```
    brouter# config t
    brouter(config)# int s0
    brouter(config-if)# ip access-group INBOUND in
```

The fourth rule is applied to the internal interface as follows:

```
    IP access-list standard OUTBOUND
        ! allow only us to go out
        permit  X.Y.Z.0  0.0.0.255
        ! all other is spoofed
        deny any log

    brouter# config t
    brouter(config)# int s0
    brouter(config-if)# ip access-group OUTBOUND out
```

Configuring the router to syslog to the log server:

```
    brouter# logging on
    brouter# logging X.Y.Z.136
```

To test if the border router is actually logging to the syslog server we change the telnet login password (had to be done anyway). Each configuration action will result in a syslog message:

```
    brouter# line vty 0 4
    brouter# login
    brouter# password <New password in plain text>
```

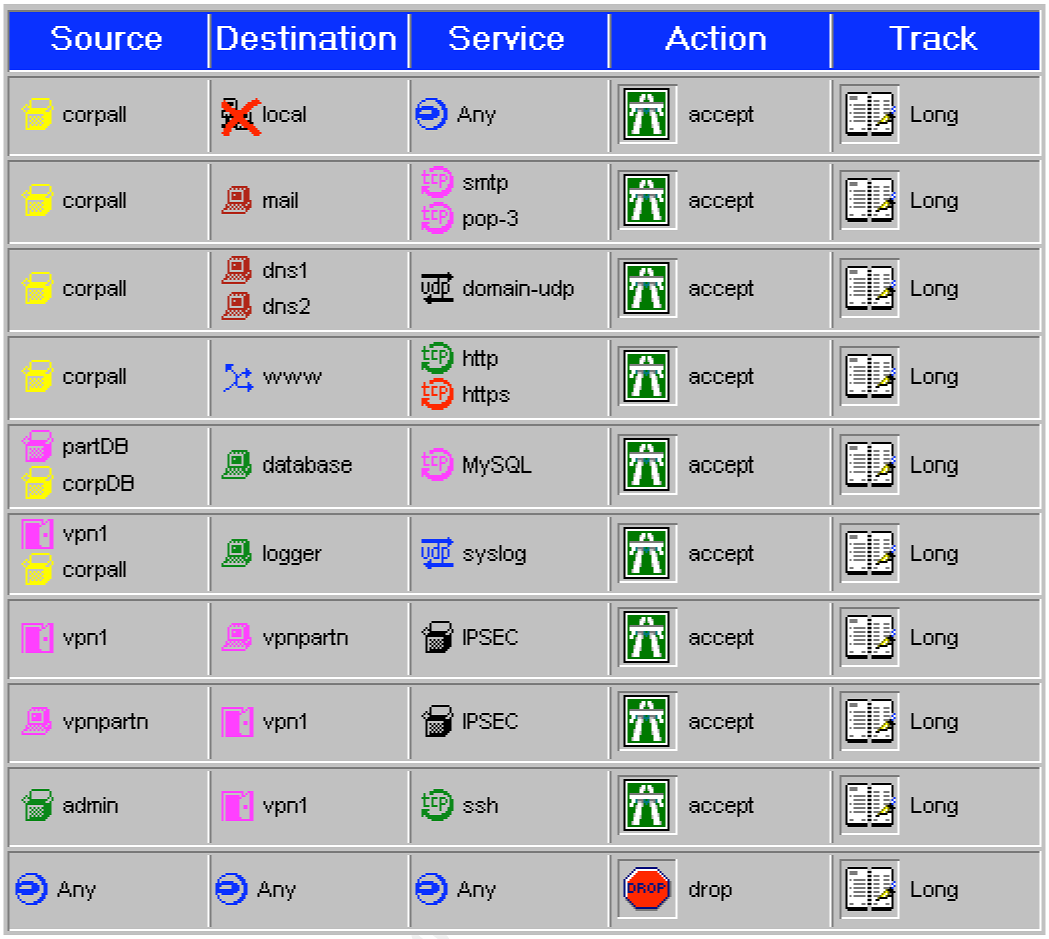
Testing the border router can be done using a packet spoofing program. There are variouspackages around which allow you to specify a source and a destination address. A spoofing tool like sirc2 can be used (author is anonymous, www.chez.com/darkweb/cf.html):

```
    testhost# gcc -Wall -o sirc2 sirc2.c
    testhost# ./sirc2 <non-local IP> X.Y.Z.140 3306
```

Above example sends a MySQL connection request to the database server from the Internet (testhost). To test if the filter actually works, the syslog server could be queried. The first rule could also be tested by connecting a local machine to a port on the outside of the border router and trying to reach another local machine, inside the border router. This would create a local packet to hit the router from the outside and should be blocked. The same kind of trick would work for the fourth rule: give an internal machine a non-local IP address and connect it on the segment between the border router and the firewall. Try to reach another non-local machine, the router should drop and log it. Again the same trick could be done for private addresses. For the 127 network, a packet spoofer program must be used since hosts won't route 127 packets to a network interface.

5. Internal firewall

The internal firewall is also a Checkpoint Firewall 1. Configuring of this rulebase is exactly the same as for the main firewall (except for different rules, of course). This rulebase is actually a lot simpler since there are no conflicting rules. Everything is quite straightforward and there is no order dependency (except for the default deny rule at the end). I'll give the full rulebase and make some comments and some testing recommendations for it.

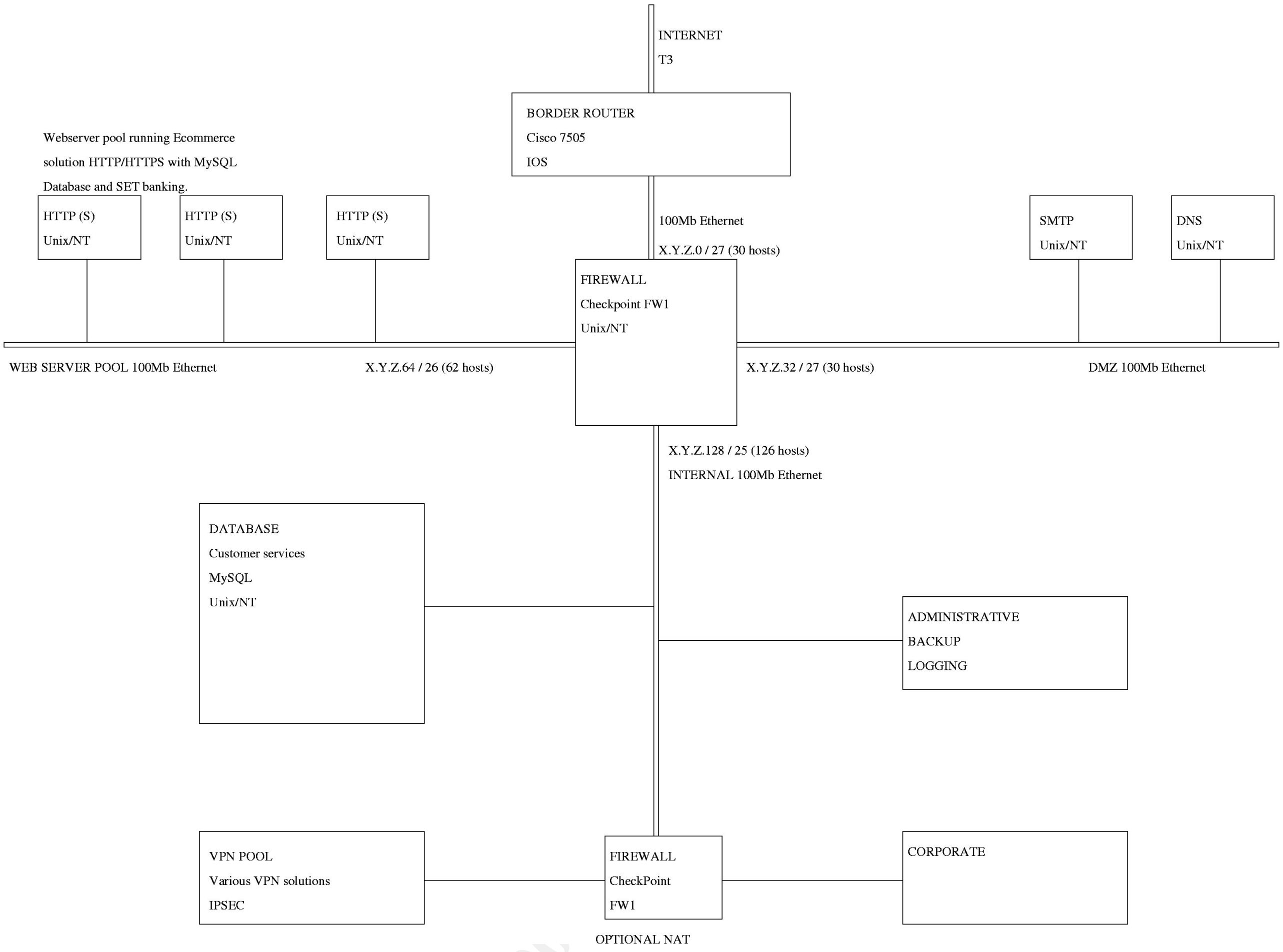


Source	Destination	Service	Action	Track
corpall	local	Any	accept	Long
corpall	mail	smtp pop-3	accept	Long
corpall	dns1 dns2	domain-udp	accept	Long
corpall	www	http https	accept	Long
partDB corpDB	database	MySQL	accept	Long
vpn1 corpall	logger	syslog	accept	Long
vpn1	vpnpartn	IPSEC	accept	Long
vpnpartn	vpn1	IPSEC	accept	Long
admin	vpn1	ssh	accept	Long
Any	Any	Any	drop	Long

It had already been remarked that the partner machines (routed through vpn1) can only enter our network through the VPN tunnel. All machines on the network have been configured to route the packets for the business partner machines through vpn1. If a packet from the business partner arrives at the main firewall, it will be dropped by the default deny rule. This is because all the rules for the partner machines should be in this (internal firewall) rulebase. This SHOULD be tested by spoofing some packets with the source IP of the business partner on the segment between the border router and the main firewall, with destination the customer database. They should be dropped by the main firewall.

Furthermore, all connections from the corporate network should be tested in the way indicated at the beginning of the main firewall rulebase.

6. Network Graph


INTERNET
T3
BORDER ROUTER
Cisco 7505
IOS
Webserver pool running Ecommerce
solution HTTP/HTTPS with MySQL
Database and SET banking.
HTTP (S)
Unix/NT
HTTP (S)
Unix/NT
HTTP (S)
Unix/NT
100Mb Ethernet
X.Y.Z.0 / 27 (30 hosts)
SMTP
Unix/NT
DNS
Unix/NT
FIREWALL
Checkpoint FW1
Unix/NT
WEB SERVER POOL 100Mb Ethernet
X.Y.Z.64 / 26 (62 hosts)
X.Y.Z.32 / 27 (30 hosts)
DMZ 100Mb Ethernet
X.Y.Z.128 / 25 (126 hosts)
INTERNAL 100Mb Ethernet
DATABASE
Customer services
MySQL
Unix/NT
ADMINISTRATIVE
BACKUP
LOGGING
VPN POOL
Various VPN solutions
IPSEC
FIREWALL
CheckPoint
FW1
CORPORATE
OPTIONAL NAT

Vincent Berk

ISTS / Dartmouth College
45 Lyme Rd., Suite 200
Hanover, NH 03755
603 646 07476

Track 2: Firewalls, Perimeter Protection and Virtual Private Networks

Assignment part 3

11/22/00

1. Assignment, planning of the assessment

The assignment is to assess the perimeter security implementation of the previous two assignments. This assignment will be done in three parts; the first part describes what will be assessed and how. The second part will be the actual assessment, including the technical details of how each test is implemented. The third part will make suggestions, based on the above assessment, on how to improve the security architecture. The tests are not exhaustive and can't be, since there are so many ways in which attackers can enter a network. Yet all the basic security will be audited.

The first three sections:

a. Verify the functionality of the firewall rulebases
b. Verify the functionality of the border router
c. Check for possibility of improper connections

These sections are all based on testing the filtering on Transport Layer and Internet layer (layer 2 and 3) of the TCP/IP model. Application level attacks can get through.

The last three sections:

d. Check the integrity of the administrator/user passwords and access to the database
e. Various other verifications
f. Check a selection of known vulnerabilities.

These sections are more aimed at the other aspects of network security. Both application level vulnerabilities as well as human factors are considered here. Users tend to choose easy to remember passwords. Applications might have bugs, which need to be patched. Those points are at least as important as the packet and protocol level filtering.

Costs will be measured in effective working hours.

a. Verify the functionality of the firewall rulebases

Because the firewalls are the main line of defense, their rulebases should be sound. This section only focuses on proper functionality of the firewalls and their logging ability. Each rule in the rulebase is tested for correct functionality. Checking for security holes and possibility of undesired access from untrusted parts of the Internet is done in section C. Yet the default-deny rule will be tested using some invalid connections, specifically using IP addresses from the business partner, which are in VPN tunnel range. This is because these packets will be routed through the VPN tunnel and thus they are denied at the main firewall.

So for each rule in the rulebase functionality has to be verified. This also means that the connection shows up properly in the log files of both the firewall and the logging server.

For each rule one or more connections are made, from the source to the destination, using the specified service and access (or no access) is checked and verified. Both firewall and logging server log files are checked.

For these tests to be successful, one host each firewall segment should be made available (or at least partly available). Furthermore, a host out on the Internet is needed (non-local IP address). OS is Unix. Access to the firewall console and log server (both to verify logs) is required. With two firewall rulebases to audit the estimate would be between 5 and 8 hours. Since there is no heavy load involved in any of these tests, they can be done safely during the day. Might problems show up in the rulebase, debugging may be required, which adds extra time to the estimate.

b. Verify the functionality of the border router

The same idea here as with the firewall. The rulebase will be checked for functionality and logging ability. Specific testing for unwanted holes in the rulebase will follow in later sections. This section is primarily to verify proper functionality of the device. Unwanted connections and password security of the border router are also assessed in later sections.

A host is needed in front and behind the border router. The host on the segment behind the border router should have a local IP address and a Unix OS. The host on the outside of the border router could be an arbitrary hosts on the Internet to which we have administrative access. In this test mainly a lot of spoofing will be done. Access to router log files and log server will be required. Debugging will be hard since it will be very hard to dump the traffic on the outside interface of the router (connection to ISP). Tests should take no more than 2 hours. There are no serious risks involved here so tests could be conducted during regular working hours.

c. Check for possibility of improper connections

This section is dedicated to unwanted finding holes in the perimeter. This will involve checking if important machines and servers are reachable from outside the perimeter, or from other unwanted locations. Also, checking needs to be done if reachable machines (like the web server) are only reachable for the services they provide to a certain group. This means that a web server, reachable by everyone on the Internet is only reachable for the HTTP and HTTPS protocols and not on any other services (ports). A DNS zone transfer will be attempted (zone transfers are used to transfer information about all the machines in a whole domain in one single query).

This will involve scanning a significant part of the address range from various locations inside and outside the local network. The scan will be performed for a large portion of the known services for the TCP and UDP protocols. ICMP requests will be tested and blocking will be verified. Specifically connecting to the database, the firewalls and the border router will from the outside, will be tested (control connections to the firewall and border router should be closed shut.)

This will be a big set of tests. Once again several hosts are needed on all the sections of the network and once again a host, to which we have administrative access, out on the Internet will be required. From these hosts ping sweeps and network scans will be performed to determine what connections are possibly open. Control connections to the firewalls and the border router will be attempted; DNS zone transfers from the Internet will be tried, as well as connections from the corporate network to the database server. Once again, access to all log files will be required. Test will very likely take up several days (but no more than 24 hours). If problems or security gaps are encountered, debugging may require extra time. Scans may generate a lot of network traffic. Due to this it would be advisable to do the tests early morning or during the night.

This section is focused on piercing the perimeter.

d. Check the integrity of the administrator/user passwords and access to the database

This section is aimed at verifying user access to data and/or administrative access to servers. It will be assured that users have unique user ID's and passwords are not easy to guess. A dictionary attack will be used on some passwords files. Also, user privileges to the customer database will be verified (which users have read-only access, which users have read-write access, is this access really required). Furthermore, the router will be checked on standard passwords (most routers come with vendor supplied, standard passwords).

For the dictionary attack on password files several password files from various machines are taken, e.g. corp1DB, corp2DB and corp3DB. (Note that not all passwords are tested. Administrators should be aware of the dangers of weak passwords. A program like npasswd could be used to enforce the use of strong passwords: www.utexas.edu/cc/unix/software/npasswd/). As a password guessing program 'Crack' will be used. Also, all vendor-supplied passwords will be tested on the border router.

Since the customer database is probably the most valuable stack of information in the company, user access has to be restricted as tight as possible. Access to the database server (both internal and business partner) will be reviewed with the management of the company. It will be ensured that all workstations are running either Unix or Windows NT to make sure all users have a unique ID. This section requires at most 8 hours. Tests can be done any time. The password cracking should be done on a non-production machine.

e. Various other verifications

This section will verify that application and security patches are up to date, virus scanners are current and whether the encrypted channels are really encrypted. If these points are not in proper working order, a strong and secure perimeter will be useless. Vulnerabilities due to weaknesses in old software will eventually be exploited and give access to the attacker right through the firewall. If data is not encrypted properly it will be readable when intercepted.

The checking of version numbers and dates will take approximately 4 hours. System administrators will provide assistance to find out which patches have been applied and what has been updated. If uncertainty arises, all changes are logged (see security policy) so log files can be reviewed. To check the encryption on the channels (HTTPS, SET and IPSEC) network dumps can be made. The configuration of the encryption/decryption software will be reviewed for correctness. This should take no more than 4 hours. A total of 8 hours for this section. No risks for the network involved. Tests can be done anytime.

f. Check a selection of known vulnerabilities

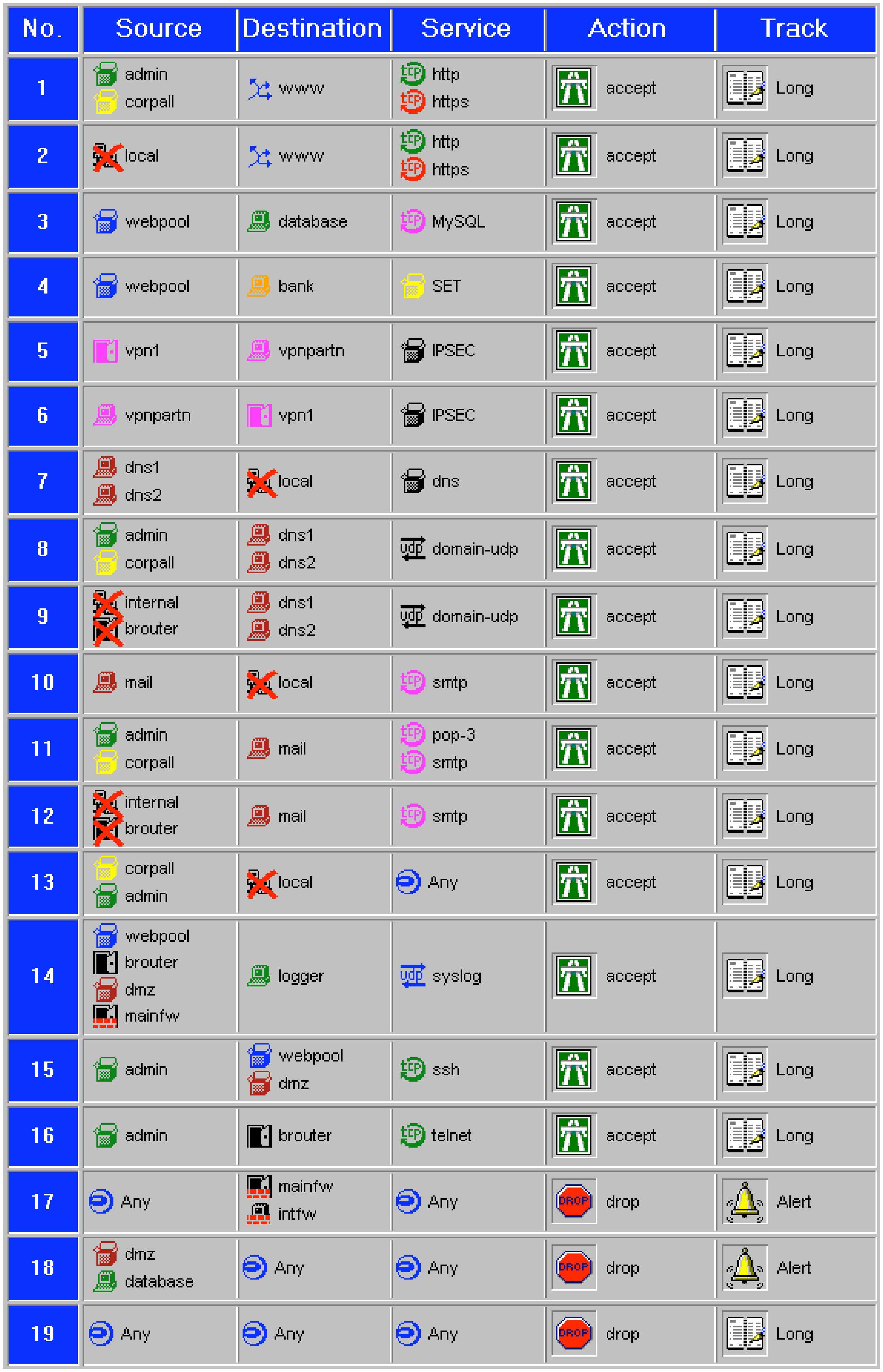
This section will aim to launch some well-known attacks on the network. From outside the perimeter various exploits are tried to see how well the security architecture stands up to all day average attacks. What will be tested if whether or not the attack succeeds and if it shows up in the log.

This is a very risky part of the assessment. Attacks and exploits should be run only under the best-controlled conditions. At least one hosts on the Internet is required to which we have administrative access. Access to all log files is required. A backup plan is needed in case something goes wrong. Before these tests are done, a backup of the entire network is made. This section will take at least 24 hours. System administrators should be present to assist if anything goes wrong. This company is running 24 hours a day.

2. Implementation of the assessment.

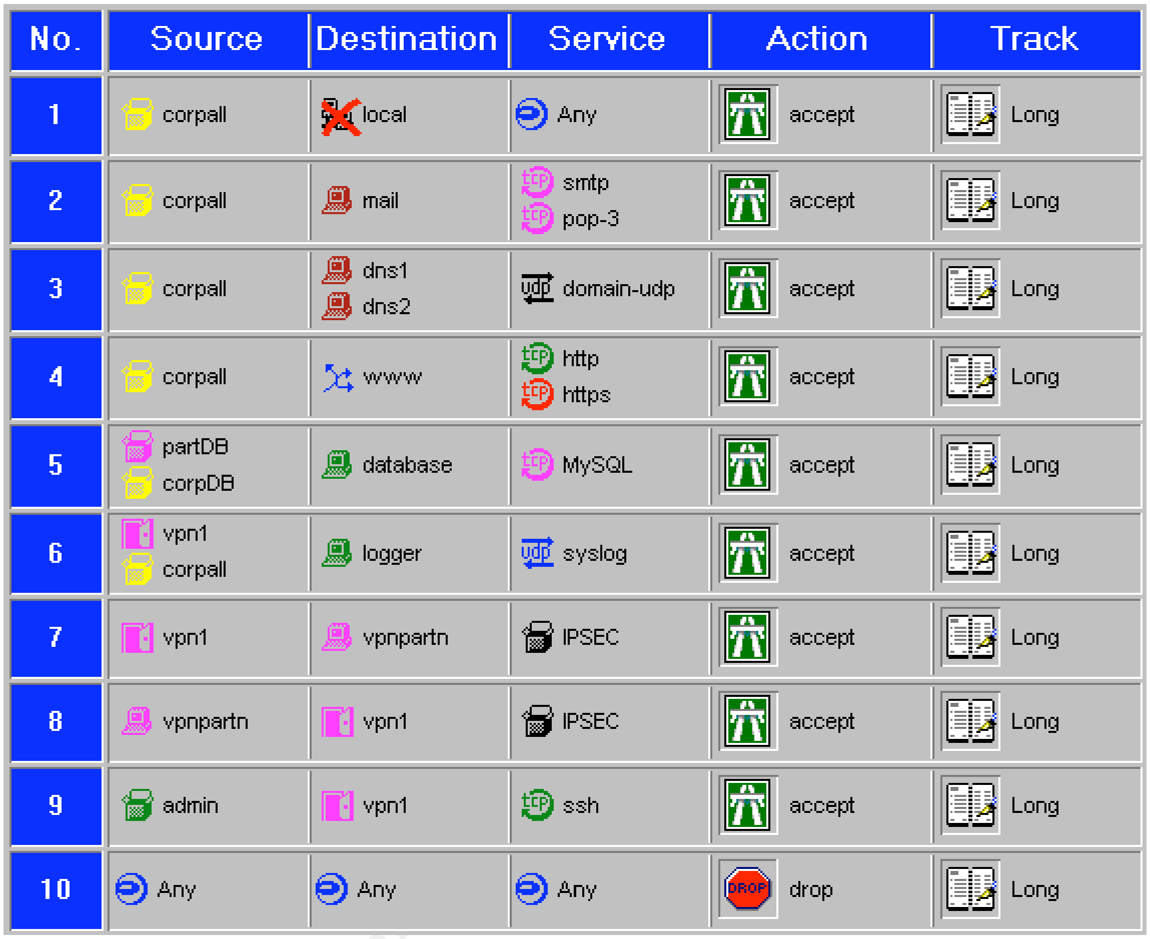
a. Verify the functionality of the firewall rulebases

```
Main firewall rulebase:
```



No.	Source	Destination	Service	Action	Track
1	admin corpall	www	http https	accept	Long
2	local	www	http https	accept	Long
3	webpool	database	MySQL	accept	Long
4	webpool	bank	SET	accept	Long
5	vpn1	vpnpartn	IPSEC	accept	Long
6	vpnpartn	vpn1	IPSEC	accept	Long
7	dns1 dns2	local	dns	accept	Long
8	admin corpall	dns1 dns2	domain-udp	accept	Long
9	internal brouter	dns1 dns2	domain-udp	accept	Long
10	mail	local	smtp	accept	Long
11	admin corpall	mail	pop-3 smtp	accept	Long
12	internal brouter	mail	smtp	accept	Long
13	corpall admin	local	Any	accept	Long
14	webpool brouter dmz mainfw	logger	syslog	accept	Long
15	admin	webpool dmz	ssh	accept	Long
16	admin	brouter	telnet	accept	Long
17	Any	mainfw intfw	Any	drop	Alert
18	dmz database	Any	Any	drop	Alert
19	Any	Any	Any	drop	Long

Internal firewall rulebase:



No.	Source	Destination	Service	Action	Track
1	corpall	local	Any	accept	Long
2	corpall	mail	smtp pop-3	accept	Long
3	corpall	dns1 dns2	domain-udp	accept	Long
4	corpall	www	http https	accept	Long
5	partDB corpDB	database	MySQL	accept	Long
6	vpn1 corpall	logger	syslog	accept	Long
7	vpn1	vpnpartn	IPSEC	accept	Long
8	vpnpartn	vpn1	IPSEC	accept	Long
9	admin	vpn1	ssh	accept	Long
10	Any	Any	Any	drop	Long

Root access is required for the machine out on the Internet. The IP address should be non-local (X.Y.Z.0/24) and not in the address range of the business partner (since their addresses will be routed through vpn1). It is also possible to set up a host on the segment between the border router and the firewall and spoof a non-local address, but then return packets will not go to this host. Network dump could be used to see if SYN gets a SYN/ACK back (to check if a connection would get established.)

I will start off with the rulebase of the main firewall. In these tests also connections through the internal firewall are tested. At the end the remaining connections of the internal firewall are verified. Note that for a connection to be established, return packets should be let through also. The log files will show if a connection was established.

The first two rules of the firewall allow all non-local, the admin machine group and the corpall machine group to reach the www logical server for HTTP and HTTPS. This is tested by connecting to the website (www.giacfortunes.com) and requesting some HTTP and HTTPS pages (this is done using a normal web browser). Connections should be made from machines in all three groups. Next the log files should be verified if the connection got logged on the firewalls and log server.

Another method would be to use netcat on the HTTP and HTTPS port and check if a connection is established (from the Internet host, a admin host and a corpall host):

```
    nc www 80
    nc www 443
```

To verify what traffic is actually on the wire use:

```
    tcpdump host www port 80
    tcpdump host www port 443
```

Again logs should be checked if the connection was successful and logged properly. Note that this also tests the fourth rule of the rulebase of the internal database for connections from corpall to www.

The next rule allows the webpool to connect to the database server using the MySQL protocol. From all web server machines a connection to the database server should be made to verify correct functionality. Log files on the log server and the firewall should be checked to verify the connection was logged and successful. Again this can be done in two ways. Having the web server connect by making manual MySQL requests or using netcat:

```
    mysql -h database -u USERNAME -p
    nc database 3306
```

Connections from the webpool to the financial service provider through the Secure Electronic Transmission protocol (SET) should be tested using the e-commerce software. Again, netcat could be used, but that could upset their machines and lock out the local web servers. Log files should be verified if the connection succeed and was logged accordingly. Again, all web server machines need to be checked.

It should be possible to initiate the VPN tunnel from both sides and should thus be tested from both sides. The full IPSEC authentication procedure should be completed from both sides to ensure that the connection can be made. Log files should once again be verified to check that the connection was successfully established and logged. Again, two rules from the internal firewall are tested too. Both the internal and the main firewall implement these two rules in their rulebase. Important is to check both firewalls and the log server for connection logs.

Both DNS servers are allowed to make DNS (both TCP and UDP) connections to all non-local machines. This is something that can be tested with netcat, or with nslookup. From the both DNS servers do:

```
    dns1# nslookup <non-local hostname>           Test UDP/53
```

```
dns1# nc -u <non-local DNS server> 53        Test UDP/53
dns1# nc <non-local DNS server> 53           Test TCP/53


dns2# nslookup                               nslookup in interactive mode
Default Server: dns1.giacfortunes.com        Shows default DNS server
Address: X.Y.Z.35                            and the ip address

> server <non-local DNS server>              Use a nonlocal server
> set type=ANY                               Try a zone transfer
> ls -d <non-local domain name>              Get the zone
```

This might fail since every capable DNS administrator will turn of zone transfers to the outside world, yet our local log files will show if the connections were allowed to go outbound. (So the zone transfer doesn't have to succeed in order for us to check if the connection was allowed by our firewall.)

Again, in all cases, if there is any uncertainty as to what is going on, on the network, tcpdump can be used to sniff all passing packets. Keep in mind that if a host is used to run tcpdump from, that host should be connected to a port on the switch that allows all traffic in the switch to be monitored.

Next are the DNS query connections (2 rules) from the admin machines, the corpall machines, the non-internal (webpool, firewall, mail, DNS servers except border router). So, connections are verified for the machines in the webpool, the admin group, the corpall group, the main firewall, the mail server (should not show up in logs since it is on the same physical segment as the DNS servers) and hosts on the Internet (the test machine). Internal firewall rule three is tested for the connections from corpall to both DNS servers. Tests with netcat (nslookup and nc can be used interchangeably):

```
web1# nc -u dns2 53         Shows up on logserver and main firewall
web3# nslookup web2         Idem.
admin1# nc -u dns1 53       Idem.
logger# nslookup mail       Idem.
corp2DB# nc -u dns2 53      Logserver, main firewall and internal firewall logs
mail# nslookup dns2         No logging (except for local logs of dns1 server)
testhost# nc -u dns1.giacfortunes.com 53      Main firewall and logserver
```

All above connections should succeed and be logged as shown. I omitted the test for the main firewall since the logs should already indicate if the main firewall has DNS connections. If the main firewall has DNS, the logs show resolved names, if not the logs show IP addresses.

Next rule allows the mail server to connect to anywhere but the local addresses. This is for SMTP connection so that mail can be delivered. Logging on both main firewall and log host:

`mail# nc testhost 25`	SMTP connection should succeed

Mail server connections are next. The same group as for the DNS servers should be able to connect to the mail server using protocol SMTP (that is: the webpool, the admin group, the corpall group, the main firewall, and hosts on the Internet: the test machine). Connection example:

`web2# nc mail 25`	Logged on firewall and log server.

Again a rule from the internal database is tested, too (nr. 2). Furthermore, the corpall group and the admin group are also allowed to make pop-3 connections to the mail server, test (and check logs accordingly):

`admin3# nc mail 110`	Logged on main firewall and log server.
`corp5# nc mail 110`	Logged on main firewall, log server and internal firewall

The next rule allows the corpall group and the admin group to connect out to the Internet (non-local addresses) without service restrictions. For the corpall group this also tests the internal firewall rule 1. Logging occurs on the main firewall and log server and, for the corpall group, also on the internal firewall:

`admin2# nc testhost 22`	SSH connection to testhost on the Internet
`corp1DB# nc testhost 80`	HTTP connection to testhost on the Internet
`admin1# nmap testhost`	Standard TCP portscan of testhost

The following rule allows the webpool, the border router, all machines in the DMZ and the main firewall to reach the syslog server for syslog (UDP/514) connections. The fact that the firewall is able to syslog to the log server has already been proved by the fact that since the first rule we've been checking the log server for log entries from the main firewall. Problems with firewall logging would have shown up there. Attempt syslog connections from the DMZ machines and the webpool:

`web1# nc -u logger 514`	Shows up in main firewall and log server logs
`mail# nc -u logger 514`	Idem.
`dns1# nc -u logger 514`	Idem.

Making a configuration change to the router will force a log entry:

```
brouter#config t
Enter configureation commands, one per line. End with CNTL/Z.
brouter(config)#^Z
brouter#
```

Router log entries can be showed on the router with the following command:

```
    brouter#show log
```

And of course the syslog server should show the same messages.

Administrative connections from the admin group to the webpool and DMZ machines (through SSH) and the border router (telnet) are the next two rules in the rulebase. They can be checked by either making a full connection with SSH/telnet or, again, using netcat:

```
    admin1# ssh root@mail
    admin1# telnet brouter
    admin1# nc dns1 22         Should drop the SSH welcome header to the terminal
```

All connections will be logged on the firewall and the syslog server.

The next two rules are alert rules. They are designed to drop the incoming packet and give a warning (according to firewall configuration). Also the log files of the firewall and the log server should show the event. Connections to the firewalls are alerted and connections initiated by the DMZ machines (other than specifically allowed by above rules) and the database are disallowed. The following connection attempts should all fail and alert:

```
    testhost# telnet X.Y.Z.2    telnet connection to the main firewall from outside
    testhost# telnet X.Y.Z.151  telnet to an interface on the internal firewall
    dns1# nc database 3306      MySQL connection from dns1 to the database
    database# nc web1 80        database server should not request webpages
    mail# ssh testhost          only SMTP connections allowed outbound
```

Since the business partner has several machines that reach the companies network through the VPN tunnel (and have database access), it is important that they actually come in through this tunnel. Therefore some packets from this range should be spoofed in front of the firewall, going inbound to the database. They should be denied by the default deny rule of the main firewall. A spoofing tool like sirc2 can be used (author is anonymous, www.chez.com/darkweb/cf.html):

```
    testhost# gcc -Wall -o sirc2 sirc2.c
    testhost# ./sirc2 <partner IP> X.Y.Z.140 3306
```

This compiles the C code and connects to the database on port 3306, spoofing the source address so the packets appear to have originated from the partners IP range (at least the partner machines which are routed through the VPN tunnel). This should be denied. Both firewall and logging server logs should indicate this.

That is enough for the main firewall. There are still some rules left in the internal firewall, which need to be verified. First of all, rule 5 has not been tested yet; both corpDB and partDB should be able to reach the database on MySQL:

```
    corp1DB# mysql -h database -u <username> -p
    part1DB# mysql -h database.giac.com -u <username> -p
```

Both should succeed and be logged by the internal firewall and the log server.

Next, both vpn1 and corpall need to be able to reach the log server on UDP/514:

```
    corp5# nc -u logger 514
    vpn1# nc -u logger 514
```

Finally, the admin machines should be able to make a SSH connection to the vpn1 machine:

```
    admin1# ssh vpn1
```

Connection should show up in the log of the internal firewall and the syslog server. All other connections should be denied. Section C will go deeper into that, yet I will give an example of a denied connection here:

```
    corp4# mysql -h database -u <username> -p
```

It is not possible to connect to the database from corp4. This packet is dropped and logged.

b. Verify the functionality of the border router

The border router is supposed to block inbound packets coming from:

Private addresses:

```
    10.0.0.0    0.255.255.255
    172.16.0.0  0.15.255.255
    192.168.0.0 0.0.255.255
    127.0.0.0   0.255.255.255
    224.0.0.0   7.255.255.255
```

Local addresses:

```
    X.Y.Z.0     0.0.0.255
```

To test this, packets should be send inbound (e.g., to interface X.Y.Z.1) with spoofed addresses from the above ranges. Again, I'll use the sirc2.c package downloaded above. Using our Internet test host (e.g. sending to SSH port):

```
testhost# ./sirc2 10.1.2.3    X.Y.Z.1  22
testhost# ./sirc2 172.16.4.5  X.Y.Z.1  22
testhost# ./sirc2 192.168.7.8 X.Y.Z.1  22
testhost# ./sirc2 127.0.0.1   X.Y.Z.1  22
testhost# ./sirc2 224.9.10.11 X.Y.Z.1  22
testhost# ./sirc2 X.Y.Z.141   X.Y.Z.1  22
```

All above packets should be denied by the border router and logged to the syslog server. Also the router has a small logging buffer that can be verified with the command:

```
brouter# show log
```

Furthermore, source routing is denied. Netcat provides us with the ability to source route a packet:

```
testhost# nc -g X.Y.Z.1 -g X.Y.Z.129 X.Y.Z.140 3306
```

Above command will try to connect to the MySQL port of the database using a source-routed packet. The source routed hops are through the internal interface of the border router and the internal interface of the main firewall. The packet should be dropped and logged.

Since we do not want to be used in a Distributed Denial of Service attack (internal hosts used to generate spoofed packets), we do not allow any spoofed packets to leave the network (permit X.Y.Z.0 0.0.0.255, deny any). The egress filter checks if all the outbound packets have the correct (a local) IP address. This is tested by placing a host between the border router and the main firewall (spoofhost). Command:

```
spoofhost# ./sirc2 123.45.67.89 testhost 23
```

This sends a packet with source address 123.45.67.89 to our Internet testhost on port 23. Packet should be dropped by the egress filter of the border router and logged to syslog.

c. Check for the possibility of improper connections

All of the following tests should fail. The network sweep scans should be performed at a time when bandwidth use is low (late at night or early morning). First, the test host on the Internet is used to scan the entire network:

```
testhost# nmap -sT -p 1- -P0 X.Y.Z.0/24
```

For each reachable host a summary is displayed:

```
Interesting ports on mail.giacfortunes.com (X.Y.Z.34)
Port    State       Protocol  Service
```

```
    25      open        tcp        smtp

    No ports open for host dns1.giacfortunes.com (X.Y.Z.35)
```

This will do a TCP scan, without pinging the hosts (ICMP echo request) for ports 1-65535 on the whole network (X.Y.Z.0/24). If any port seems open while it should have been closed, there is a security risk. The same can be done for UDP, although that takes a long time

```
    testhost# nmap -sU -p 1- -P0 X.Y.Z.0/24

    No ports open for host mail.giacfortunes.com (X.Y.Z.35)

    Interesting ports on dns1.giacfortunes.com (X.Y.Z.34)
    Port    State       Protocol  Service
    53      open        udp        domain
```

Both scans should be tried from the corporate network and the business partners network (if they agree to this, of course):

```
    corp1DB# nmap -sT -p 1- -P0 X.Y.Z.0/24
```

Log files will scream of thousands of blocked connection attempts. That is now it should be. A zone transfer from the DNS servers should fail:

```
    testhost# nslookup
    Default Server: dns1.giacfortunes.com
    Address: X.Y.Z.35

    > server dns2.giacfortunes.com
    > set type=ANY
    > ls -d giacfortunes.com
```

Connection failures will all be logged on the main firewall and the syslog server. Connection attempts to the main firewall and the internal firewall have already been done in a previous section. A very simple ping sweep can be done by nmap too:

```
    testhost# nmap -sP -PI X.Y.Z.0/24
```

If the output looks like this:

```
    Starting nmap V. 2.12 by Fyodor (fyodor@dhp.com,
www.insecure.org/nmap/)
    ...
    Host mail.giacfortunes.com (X.Y.Z.34) appears to be up.
    Host dns1.giacfortunes.com (X.Y.Z.35) appears to be up.
    Host dns2.giacfortunes.com (X.Y.Z.36) appears to be up.
    Host www.giacfortunes.com (X.Y.Z.66) appears to be up.
    ...
```

Then the firewall is not blocking ICMP requests (which is should be doing, according to the security policy. ICMP requests can be blocked in the firewall properties window in the policy editor).

d. Check the integrity of the administrator/user password and access to the database

First, it will be ascertained that all machines are either running a Unix flavor or Windows NT/2000. Next, it will be verified if all employees are using a unique user ID/password combination. Finally, read-only/read-write access to the customer database will be reviewed for all accounts. This work does will be done together with a member of the management.

For the password cracking several password files are taken. For this a Unix (/etc/passwd) file and a Windows NT password file are taken. To do a dictionary attack on the /etc/passwd file, Crack is used (http://packetstorm.securify.com/crackers/crack). For Windows NT l0phtcrack is used (http://www.l0pht.com/l0phtCrack).

Install Crack (see manual) and run a dictionary attack on a downloaded password file:

```
    crack /tmp/passwd.corp1DB
```

L0phtCrack has a nice GUI and is installed easily. Import the SAM password hash into the program and start cracking.

Finally, the Border Router has to be checked on standard Cisco supplied passwords. This is simply done by telnetting to the router and trying them out.

```
    admin1# telnet brouter
```

e. Various other verifications

Checking security patches, software updates, virus scan versions, encrypted channel configuration is all done by hand with the help of the system administrators. Yet a network dump can be made to check if the information flowing through the HTTPS, SET and IPSEC connections is actually encrypted. A host is used on the segment between the border router and the main firewall (for convenience, the same host is used as before: spoofhost):

```
    spoofhost# tcpdump -s 1600 host web1 and host bank
    spoofhost# tcpdump -s 1600 host vpn1 and host vpnpartn
```

Above are two examples of how to get tcpdump to output the whole packet (an Ethernet packet is only 1500 bytes so this will print all). The communication between web1 and bank and between vpn1 and vpnpartn is dumped this way. If the information looks 'random' enough, there is probably nothing wrong. This is not a very reliable test though, but as long as there is no plain text going between the hosts, it works.

f. Check a selection of known vulnerabilities

Once again: This section is not absolutely necessary and creates a huge risk for the continuity of the ongoing business.

First, a very simple attack on the web servers: double dot.
This actually means that you try to trick the web server into providing you with files that were not meant to be exported. This works by putting a relative path in the HTTP request:

```
    testhost# cat httpddhead
    GET /../../../../etc/passwd HTTP/1.0


    testhost#
```

Send it to the webserver by:

```
    testhost# cat httpddhead | nc www 80
```

If the web server is vulnerable to this attack, the /etc/passwd file might be returned on this request. (This attack can be much more sophisticated, using Unicode etc. but I just want to get the idea clear.)

As a second attack I use the well-known teardrop attack (Bugtraq ID 124) and can be found at: http://www.securityfocus.com/bid/124 . Teardrop is a denial of service attack exploiting the way kernels rebuild fragmented IP packets. Multiple fragment packets have overlapping offsets. This attack has been around for a while and causes a whole range of different problems. The C code for the exploit can be downloaded at securityfocus. Compile with:

```
    testhost# gcc -Wall -o teardrop teardrop.c
```

Running this against several IP addresses in the webpool or DMZ should not cause any troubles (hanging or rebooting of machines). If it does, patches and updates are available.

Finally, the mailserver is checked for relaying. This can be done by connecting to the mail server on the SMTP port and trying to specify a non-local sender and a non-local receiver. The mail server will deny this if relaying is switched off. Transcript:

```
    testhost# nc mail.giacfortunes.com 25
    220 mail.giacfortunes.com ESMTP Sendmail 8.9.3/8.9.3; Tue, 14 Nov
2000 20:11:21 -0500
```

Say hi:

```
    HELO testhost
    250 mail.giacfortunes.com Hello testhost [N.M.P.Q], pleased to meet
you
```

Specify sender:

```
    MAIL FROM: <non-local address 1>
    250 <non-local address 1>... Sender ok
```

Specify receiver:

```
    RCPT TO: <non-local address 2>
    250 <non-local address 2>... Recipient ok
```

At this point it is clear that mail relaying is enabled. This means the mail server can be used to distribute enourmous quantities of SPAM. This is a bad thing.

3. Proposed Improvements

In general it can be said that the perimeter is very tight. There is very few things allowed and there are several single points of failure.

First of all, there is no way for administrators to do administrative tasks from the outside the perimeter. All inbound connections are denied. If anything needs attention, administrators have to come over and enter the building to tackle the problems. This is a security advantage but also a huge inconvenience.

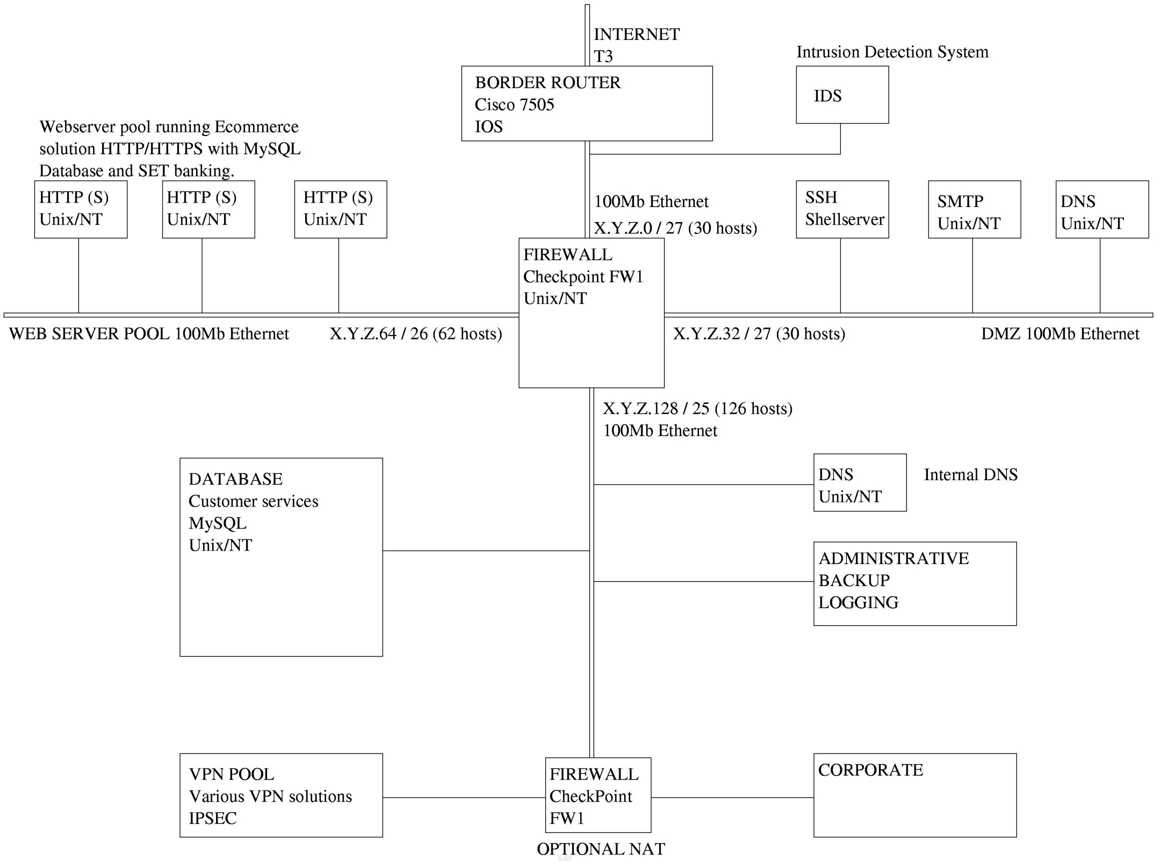
Next, there is no way in which email messages can be downloaded from outside the perimeter. The pop-3 connections are only allowed for a very tightly controlled group of machines inside the internal network. If email is to be checked, employees have to physically enter the building.

Both these inconveniences could be solved with a 'shell-server' in the DMZ. This would be a machine to which the outside world (Internet) is allowed to make SSH connections. Then from this shell-server administrators would be allowed to connect to the admin machine group. Employees could use this server to check their email from outside. The only service running on this shell-server would (of course) be SSH, to minimize the chances of a security breach.

Another issue to consider might be split DNS. Right now, the company has only one name resolving service. Even though DNS zone transfers are disabled, it is still possible to just query the DNS for each IP address and see if it comes up with a name. This way attackers can still collect information about the machines on the network.

If split DNS is used, the DNS servers in the DMZ only resolve the names for the webserver (www), the mail server (mail) and both the DNS servers (dns1 and dns2). Inside the perimeter (internal network X.Y.Z.128/35) there would be a third DNS (dns3) server to resolve the names of all the internal machines. All internal machines would then query from the internal DNS server. There would be no clue on the outside as to what machines are at the inside of the perimeter.

For added security an Intrusion Detection System (IDS) could be set up between the border router and the main firewall. This IDS could detect network scans, sweep pings, invalid packets, attack signatures etc. and warn system administrators of possibly hostile IP addresses. These hostile IP's could then (after thourough consideration) be blocked in the border router.



```
Graph of the network after adding the IDS the SSH shell-server and the
third internal DNS (split-DNS system).
```

Finally, one minor point of consideration. It is a fact that there are no mirror sites. This obviously means a single point of failure. When anything goes wrong with this site, the company won't be able to provide fortune cookies for a while. Mirror sites could be connected through VPN tunnels, since the architecture for that is already available.

Adding a third firewall, in parallel with the main firewall, could create more redundancy. This firewall could then take over part of the load of the first one, or just sit and wait until the first one fails and then take over.