Global Information Assurance Certification Paper

HOW TO DEVELOP A FIREWALL CONFIGURATION FROM YOUR SECURITY POLICY

Bernd K. Walter
SANS Security Essentials Practical
v. 1.4, Option 1

Abstract

Even if firewalls get smarter and many companies and organisations already have a firewall installed, we still see security holes due a poorly configured firewall. So why wait till the hole is found during a self issued or much more worse from a non self issued security audit ?
The intention of this paper is to discuss some issues to avoid configuration errors during firewall setup and to get a detailed documentation of the firewall configuration by manually setting up a Cisco PIX firewall.

Overview or Where We Are

Developing a firewall configuration is just a small piece in a jigsaw which is only a piece in another big jigsaw, but if this small piece not fits the whole jigsaw cannot be completed.

Based on the corporate security policy we can follow the Security wheel [1] to keep our security implementations up to date.

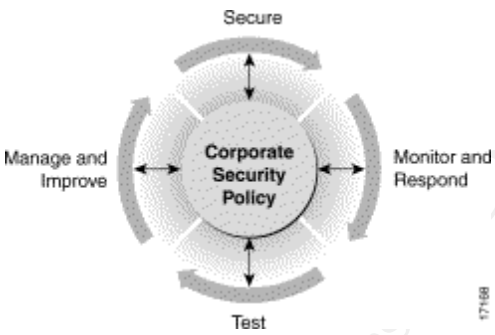

fig. 1

Our small piece of the jigsaw is part of the "secure" phase and only discuss one line of defence in our multiple line of defence security system namely the rule set of a stateful inspection firewall engine.

We will discuss these steps

- Prerequisites
  - Get a detailed map of the complete network
  - Get clear policies from your Chief Security Officer
  - Get a detailed description of what your firewall actually is able to do

- Develop the rules from the policy
  - Define network objects
  - Brief description of communications and used protocols
  - Review the Policy
- Rules Implementation (Cisco Secure Pix Firewall example)
  - Basics, IP addressing and routing
  - NAT + ACLs per service
  - Complete Firewall Configuration
- Further steps

Prerequisites

The first step to configure your firewall is to get all the basic information you need:

- Get a detailed map of the complete network
- Get clear policies from your Chief Security Officer
- Get a detailed description what your firewall actually is able to do

Get a detailed map of the complete network

To clearly understand what your Chief Security Officer (CSO) wants to be implemented and to agree to the same basics it is absolutely necessary to have a detailed map of the network.
This map not necessarily needs to contain all IP addresses; you also can group logical units and later when defining the real network objects you can add the IP addresses. Here is a very simple example for the map:

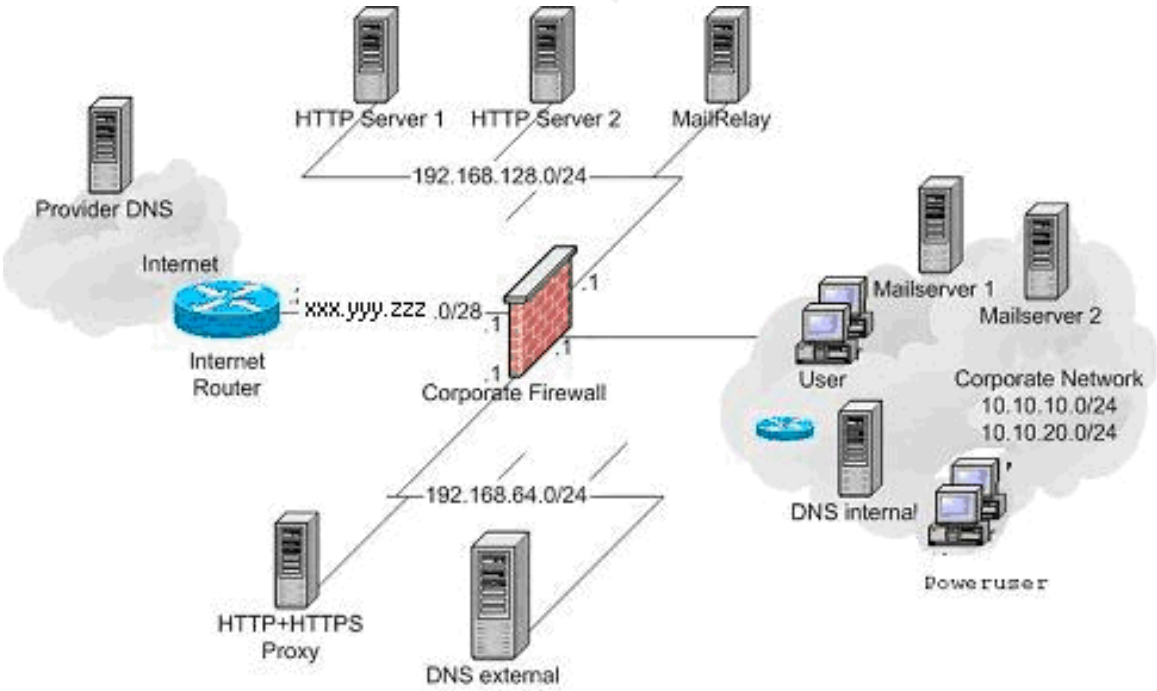

fig. 2

Get clear policies from your Chief Security Officer

The next step is to extract the necessary information from the Security Policy to configure your firewall. You can get this information from your CSO (in larger organisation this is not always the CSO, maybe there ´s a technical advisor , but in this paper I will name him CSO). Discuss all points until it is absolutely clear what the goals are, write it down in detail and get it signed by your CSO.
For instance:

The parent policy of this firewall is to deny anything. Each communication through the firewall will be inspected and only allowed when there is a corresponding rule in the policy.

Users are allowed to:

- Send and receive emails using the corporate email servers
- Connect to the internet using http and https only via the corporate proxy
- Connect to the corporate http server group using http and https

Powerusers are allowed to

- directly access the internet using http, https, ftp, dns and ssh
- Connect to the corporate http server group using http and https
- Send and receive emails using the corporate email servers

HTTP+ HTTPS proxy is allowed to

- Connect to the internet using http, https and dns

Mailservers are allowed to

- Connect to the Mailrelay using smtp
- Connect to the users

The Mailrelay is allowed to

- Connect to the internet using smtp
- Connect to the corporate Mailservers using smtp

The corporate internal DNS Server is allowed to

- Send DNS name queries to the corporate external DNS and the provider DNS

The corporate external DNS Server is allowd to

- Send dns name queries to the internet

The Provider dns server is allowed to

- Connect to the external corporate dns server

The internet is allowed to

- Connect to the http server group using http.
- Connect to the Mail Relay using smtp

Get a detailed description what your firewall actually is able to do

Another prerequisite for implementing the firewall rules is to know how the firewall works and what the limitations of the firewall are.
Here as an example the cisco PIX firewall [2]:

The PIX firewall is a stateful inspection [3] engine (called adaptive security algorithm by cisco) with stateful support for the listed protocols.
It is not an application level gateway, so filtering content has to be implemented using a 3rd party product.

- Hardware: 200MHz Prozessor, 32 MB RAM, Flash: 8 MB, 2 onboard 10/100BaseT-Fast Ethernet

TCP/IP protocol and aplication support

- Internet Protocol (IP)
- Transmission Control Protocol (TCP)
- User Datagram Protocol (UDP)
- Internet Control Message Protocol (ICMP)
- Generic Route Encapsulation (GRE)
- Address Resolution Protocol (ARP)
- Domain Name System (DNS)
- Simple Network Management Protocol (SNMP)
- Boot Protocol
- HyperText Transport Protocol (HTTP)
- File Transfer Protocol (FTP)
- Trivial File Transfer Protocol (TFTP)
- Archie
- Gopher
- Telnet
- NetBIOS über IP (Microsoft Networking)
- Point-to-Point Tunneling Protocol (PPTP)
- SQL*Net (Oracle Client/Server-Protokoll)
- Sun RPC-Dienste (Remote Procedure Call), einschließlich Network File System (NFS)

Multimedia

- Microsoft NetShow
- White Pine CU-SeeMe
- RealNetworks RealAudio und RealVideo
- Xing StreamWorks
- VDOnet VDOLive
- VXtreme WebTheater
- VocalTec Internet Phone

Videoconferencing (H.323)

- Microsoft NetMeeting
- Intel Internet Video Phone
- White Pine Meeting Point

Develop the rules from the policy

The 2nd step to setup your firewall is to deploy the rules from the given and signed policy.

- Define network objects (Remember NAT)
- Brief description of communications and used protocols
- Review the Policy

Define network objects

Based on the map of the network and the policy we can set up the network objects list containing all network objects which need to communicate through the firewall. This is also a good point to think about network address translation (NAT).
There are two big issues with NAT:
Private IP addresses (RFC 1918 [4]) are not routed through the internet so if we use these addresses in our corporate network, like in this example, we need to use NAT [5] for translating our inside addresses to outside official IP addresses.
Using dynamic NAT for non server systems would change their global IP addresses to a "moving target ", which will not prevent the client from being attacked, but making it a little harder.

So for our http servers and the mailrelay we will use static NAT because they need to be permanently reachable from the internet and for our proxy and the Powerusers we will use dynamic NAT.

The network objects list could look like this:

class	group	network object	short name	IP address	outside NAT IP	service
networks		outside segment	out	xxx.yyy.zzz.0/28		
		dmz1 segment	dmz1	192.168.128.0/24		
		dmz2 segment	dmz2	192.168.64.0/24		
		corporate range	inside	10.10.10.0/24 10.10.20.0/24		
Server	http server	http server1	www1	192.168.128.100	xxx.yyy.zzz.10	http,https
		http server2	www2	192.168.128.101	xxx.yyy.zzz.11	http,https
		http+https proxy	proxy	192.168.64.200	xxx.yyy.zzz.12	http,https
		Mail Relay	MR	192.168.128.200	xxx.yyy.zzz.13	SMTP
	Mail Server	Mailserver 1	MSVR1	10.10.10.101		SMTP
		mailserver 2	MSVR2	10.10.10.102		SMTP

		DNS external	EDNS	192.168.64.100	xxx.yyy.zzz.14	DNS
		DNS internal	IDNS	10.10.10.100	xxx.yyy.zzz.9	DNS
		DNS Provider	PDNS	194.25.0.60	194.25.0.60	DNS
Clients	Poweruser	Poweruser	PU	10.10.10.128/25	xxx.yyy.zzz.9	http, https,dns FTP
	user	User	User	10.10.20.0/24		http, https

So we have to look for the protocols HTTP, HTTPS, SMTP, FTP and DNS and how these protocols will affect our firewall rules.

Brief Description of communications and used protocols

For the description of the communications crossing he firewall we will only use the short names from the list ahead. Also we only use the destination port number for description of these connections, because most of the discussed protocols use a source port number greater than 1023 and the stateful inspection engine automatically opens the reverse ports. . If there are any exceptions we will pinpoint these in the description part.

For a reference of well known TCP and UDP port you can have a look in your /etc/services file or for the official list of assigned numbers you can have a look to IETF RFC 1700 [6]. This includes TCP/UDP well known ports as well as IP protocol numbers.

DNS

It is not the intention of this paper to discuss the world best implementation of a DNS [7+8] design. Our assumption is, there is a given DNS concept which uses an internal DNS Server (IDNS) for providing internal DNS services only for User and PU (see list above) and forwarding DNS queries to the EDNS or to the PDNS if there are queries for names outside the corporate network

The EDNS is the primary DNS for the external domain and uses the PDNS as a secondary for this domain. So the PDNS needs to initialize a DNS zone transfer to the EDNS [9].

All the communications used for DNS through the firewall are listed here

firewall interfaces /networks involved	Protocol/ Port	object and communication	remarks
inside « dmz2	udp 53	IDNS ® EDNS PU -> EDNS	Lookup
inside « out	udp 53	IDNS -> PDNS PU -> PDNS	Lookup (Backup)
dmz1 <-> dmz2	udp 53	MR -> EDNS	Lookup
dmz1 <->out	udp 53	MR -> PDNS	Lookup
dmz2 « out	udp 53	EDNS -> Internet Proxy -> PDNS	Lookup
	tcp 53	PDNS -> EDNS	Zonentransfer

Some implementations of DNS Servers still use source port 53 when talking to other nameserver so this should be considered in the configuration.

Email services (SMTP)

Again we assume a given concept. All mail traffic is handled by the corporate mailservers. If it is destined or comes from the internet it needs to be processed by the mailrelay (MR).
The only protocol supported through the firewall is SMTP.
All the communications used for SMTP [10] through the firewall are listed here

firewall interfaces /networks involved	Protocol/ Port	object and communication	remarks
inside « dmz1	tcp 25	MSVR1+2 ® MR MR -> MSVR!+2	forward
dmz1 <-> out	tcp25	MR -> Internet Internet -> MR	forward

Web based Services (http, https, ftp)

Users only reach the corporate HTTP servers (WWW1 + WWW2) directly, to connect to the internet they have to use the Proxy. They are only allowed to use the protocols http and https.

Powerusers are allowed to connect to WWW1 and WWW2 directly using http and https.
They are also allowed to connect to the internet directly using http, https and ftp.
WWW1 + WWW2 provide webservices to the internet.

All the communications used for web based services through the firewall are listed here

firewall interfaces /networks involved	Protocol/ Port	object and communication	remarks
inside <-> out	tcp 80, 443, 21	PU -> Internet	http,https, ftp
inside <-> dmz1	tcp 80, 443	User + PU -> WWW 1+2	
inside « dmz2	tcp 80, 443	user -> Proxy	http + https
dmz2 <-> out	tcp 80, 443	Proxy -> Internet	http + https
dmz1 « out	tcp 80	Internet -> WWW 1 + 2	http server

Bringing it together we get one list with the complete set of communications

traversing the firewall.
At this point we should carefully check the original policy to verify, that we are still synchronized and if correct we now can go on and establish the firewall setup.

Rules implementation

- Basics, IP addressing and routing
- DNS
- Mail
- Web based services
- Configuration file

Even if there are some smart graphical tools today to configure a firewall I will implement this policy using the command line interface (CLI) of a Cisco PIX firewall to demonstrate all the necessary steps to be taken.

Basics, IP addressing and routing

There are two major issues to know when configuring a PIX firewall. The first is, that no traffic is forwarded when there is no translationslot configured (a translation with identical local and global address still is a translation). The 2nd point is the default behaviour of the PIX when there are no security rules applied:
Traffic from a network behind an interface with a higher security level is allowed to access networks behind an interface with a lower security level.
Traffic from a lower security level will not reach networks behind a higher security level if there is not a special rule which allows this. IP packets from interfaces with identical security levels will never see each other through the PIX.

The first point is to name the PIX and set your passwords.

enable password *youre favourite password phrase, for configuration and enhanced troubleshooting*
passwd *youre 2nd favourite password phrase for basic admin and basic troubleshooting*
hostname *don´t call it pix version xx*

Then configure the security levels the speed and the IP addresses of the interfaces, from the network map. To enable an interface it ´s just the *interface ethernet0* auto command. There is no support for the shut and no shut commands from Cisco IOS.

nameif ethernet0 outside security0
nameif ethernet1 inside security100
nameif ethernet2 *dmz1 security20*
nameif ethernet3 *dmz2 security40*

interface ethernet0 auto

interface ethernet1 auto
interface ethernet2 auto
interface ethernet3 auto

ip address outside xxx.yyy.zzz.1 255.255.255.224
ip address inside 10.10.10.1 255.255.255.0
ip address dmz1 192.168.128.1 255.255.255.0
ip address dmz2 192.168.64.1.255.255.255

Now we can set up the routing. In our example we have a default route to the internet and a dedicated route to the 2nd internal network which is not directly reachable via our inside interface. There is no support for secondary IP addresses on a PIX firewall.

route outside 0.0.0.0 0.0.0.0 xxx.yyy.zzz.2
route inside 10.10.20.0 255.255.255.0 10.10.10.2

Based on the network map, the brief descriptions and the network objects, now we can setup the NAT and the access-lists for the PIX Firewall. I will do this step by step with respect to each used protocol. Doing this step by step allows recognizing errors of the configuration concept in a very early state or later helps troubleshooting.

DNS

This is the Pix firewall configuration which is directly derived from the protocol descriptions and the network objects list. Included are the NAT configuration and the appropriate access-list for enabling the corporate DNS concept. The Nat 0 command is used, because no traffic will be forwarded if there is no translation slot as mentioned before. If dynamic NAT is possible we will use it.

firewall interfaces /networks involved	Protocol/ Port	object and communication	remarks
inside « dmz2	**udp 53**	**IDNS ® EDNS PU -> EDNS**	**Lookup**
Nat 0 10.10.10.100 255.255.255.255 *Nat 0 10.10.10.128 255.255.255.128* *access-list acl_inside permit udp host 10.10.10.100 gt 1023 host 192.168.64.100 eq domain* *access-list acl_inside permit udp 10.10.10.128 255.255.255.128 gt 1023 host 192.168.64.100 eq domain*			
inside « out	**udp 53**	**IDNS -> PDNS PU -> PDNS**	**Lookup**
Global 1 xxx.yyy.zzz.9 *Nat 1 10.10.10.100* *255.255.255.255* *Nat 1 10.10.10.128* *255.255.255.128* *access-list acl_inside permit udp host 10.10.10.100 gt 1023 host 194.25.0.60 eq domain* *access-list acl_inside permit udp 10.10.10.128 255.255.255.128 gt 1023 host 194.25.0.60 eq domain*			
dmz1 <-> dmz2	**udp 53**	**MR -> EDNS**	**Lookup**
Nat 0 192.168.128.200 255.255.255.255 *access-list acl_dmz1 permit udp host 192.168.128.200 gt 1023 host 192.168.64.100 eq domain*			

dmz1 <->out	udp 53	MR -> PDNS	Lookup
static (dmz1,outside) xxx.yyy.zzz.13 192.168.128.200 netmask 255.255.255.255 0 2000 access-list acl_dmz1 permit udp host 192.168.128.200 gt 1023 host 194.25.0.60 eq domain			
dmz2 « out	**udp 53**	**Proxy -> PDNS**	**Lookup**
Global 2 xxx.yyy.zzz.12 Nat 2 192.168.64.200 255.255.255.255 access-list acl_dmz2 permit udp host 192.168.64.200 gt 1023 host 194.25.0.60 eq domain			
dmz2 « out	**udp 53**	**EDNS -> Internet**	**Lookup**
	tcp 53	**PDNS -> EDNS**	**Zonentransfer**
static (dmz2,outside) xxx.yyy.zzz.14 192.168.64.100 netmask 255.255.255.255 0 2000 access-list acl_dmz2 permit udp host 192.168.64.100 gt 1023 host 194.25.0.60 eq domain access-list acl_ouside permit tcp host 194.25.0.60 gt 1023 host xxx.yyy.zzz14 eq domain			

Mail

As you see in this configuration the mailservers and the mailrelay are statically translated because they need to be permanently reachable. The internal mailservers only talk to the mailrelay and the mailrelay communicate to the internet.
In the access-lists you see some denies. This first groups task is to prevent the internal networks from the mailrelay in case of an intruder. The second group is to prevent the mailrelay from spoofed IP addresses.

firewall interfaces /networks involved	Protocol/ Port	object and communication	remarks
inside « dmz1	**tcp 25**	**MSVR1+2 ® MR MR -> MSVR 1 +2**	**forward**
static (inside,dmz1) 10.10.10.101 10.10.10.101 netmask 255.255.255.255 0 2000 static (inside,dmz1) 10.10.10.102 10.10.10.102 netmask 255.255.255.255 0 2000 access-list acl_inside permit tcp host 10.10.10.101 gt 1023 host 192.168.128.200 eq 25 access-list acl_inside permit tcp host 10.10.10.102 gt 1023 host 192.168.128.200 eq 25 access-list acl_dmz1 permit tcp host 192.168.128.200 gt 1023 host 10.10.10.101 eq 25 access-list acl_dmz1 permit tcp host 192.168.128.200 gt 1023 host 10.10.10.102 eq 25			
dmz1 <-> out	**tcp25**	**MR -> Internet Internet -> MR**	**forward**
static (dmz1,outside) xxx.yyy.zzz.13 192.168.128.200 netmask 255.255.255.255 0 2000 access-list acl_dmz1 deny tcp host 192.168.128.200 any 10.10.10.0 255.255.255.0 eq 25 access-list acl_dmz1 deny tcp host 192.168.128.200 any 10.10.20.0 255.255.255.0 eq 25 access-list acl_dmz1 deny tcp host 192.168.128.200 any 192.168.64.0 255.255.255.0 eq 25 access-list acl_dmz1 permit tcp host 192.168.128.200 gt 1023 any eq 25 access-list acl_ouside deny tcp 10.10.10.0 255.255.255.0 host xxx.yyy.zzz.10 eq 25 access-list acl_ouside deny tcp 10.10.20.0 255.255.255.0 host xxx.yyy.zzz.10 eq 25 access-list acl_ouside deny tcp 192.168.128.0 255.255.255.0 host xxx.yyy.zzz.10 eq 25 access-list acl_ouside deny tcp 192.168.64.0 255.255.255.0 host xxx.yyy.zzz.11 eq 25 access-list acl_outside permit tcp any gt 1023 host xxx.yyy.zzz.13 eq 25			

Web based services

So here is the list of the web based services. Maybe you wonder about the *0 2000* in the static commands. These two fields limit the maximum number of established sessions (the first number, 0 = unlimited) and of embryonic sessions (second number) used with this translation. This will help to prevent the Servers from crashing during a SYN-Flood attack. This number should be adjusted regarding to the amount of expected sessions to the server.

firewall interfaces /networks involved	Protocol/ Port	object and communication	remarks
inside <-> out	**tcp 80, 443, 21**	**PU -> Internet**	**http,https, ftp**
Global 1 xxx.yyy.zzz.9 Nat 1 10.10.10.128 255.255.255.128 Access-list acl_inside deny tcp 10.10.10.128 255.255.255.128 192.168.128.0 255.255.255.0 eq 80 Access-list acl_inside deny tcp 10.10.10.128 255.255.255.128 192.168.128.0 255.255.255.0 eq 443 Access-list acl_inside deny tcp 10.10.10.128 255.255.255.128 192.168.128.0 255.255.255.0 eq 21 Access-list acl_inside deny tcp 10.10.10.128 255.255.255.128 192.168.64.0 255.255.255.0 eq 80 Access-list acl_inside deny tcp 10.10.10.128 255.255.255.128 192.168.64.0 255.255.255.0 eq 443 Access-list acl_inside deny tcp 10.10.10.128 255.255.255.128 192.168.64.0 255.255.255.0 eq 21 access-list acl_inside permit tcp 10.10.10.128 255.255.255.128 gt 1023 any eq 80 access-list acl_inside permit tcp 10.10.10.128 255.255.255.128 gt 1023 any eq 443 access-list acl_inside permit tcp 10.10.10.128 255.255.255.128 gt 1023 any eq 21			
inside <-> dmz1	**tcp 80, 443**	**User + PU -> WWW 1+2**	**http + https**
Nat 0 10.10.10.128 255.255.255.128 Nat 0 10.10.20.0 255.255.255.0 access-list acl_inside permit tcp 10.10.10.128 255.255.255.128 gt 1023 host 192.168.128.101 eq 80 access-list acl_inside permit tcp 10.10.10.128 255.255.255.128 gt 1023 host 192.168.128.102 eq 80 access-list acl_inside permit tcp 10.10.10.128 255.255.255.128 gt 1023 host 192.168.128.101 eq 443 access-list acl_inside permit tcp 10.10.10.128 255.255.255.128 gt 1023 host 192.168.128.102 eq 443 access-list acl_inside permit tcp 10.10.20.0 255.255.255.0 gt 1023 host 192.168.128.101 eq 80 access-list acl_inside permit tcp 10.10.20.0 255.255.255.0 gt 1023 host 192.168.128.102 eq 443			
inside « dmz2	**tcp 80, 443**	**user -> Proxy**	**http + https**
Nat 0 10.10.20.0 255.255.255.0 access-list acl_inside permit tcp 10.10.10.128 255.255.255.128 gt 1023 host 192.168.64.200 eq 80			
dmz2 <-> out	**tcp 80, 443**	**Proxy -> Internet**	**http + https**

Global 2 xxx.yyy.zzz.12			
Nat 2 192.168.64.200 255.255.255.255			
Access-list acl_dmz2 deny tcp host 192.168.64.200 192.168.128.0 255.255.255.0 eq 80			
Access-list acl_dmz2 deny tcp host 192.168.64.200 192.168.128.0 255.255.255.0 eq 443			
Access-list acl_dmz2 deny tcp host 192.168.64.200 10.10.10.0 255.255.255.0 eq 80			
Access-list acl_dmz2 deny tcp host 192.168.64.200 10.10.10.0 255.255.255.0 eq 443			
Access-list acl_dmz2 deny tcp host 192.168.64.200 10.10.20.0 255.255.255.0 eq 80			
Access-list acl_dmz2 deny tcp host 192.168.64.200 10.10.20.0 255.255.255.0 eq 443			
access-list acl_dmz2 permit tcp host 192.168.64.200 gt 1023 any eq 80			
access-list acl_dmz2 permit tcp host 192.168.64.200 gt 1023 any eq 443			
dmz1 « out	**tcp 80**	**Internet -> WWW 1 + 2**	**http server**
static (dmz,outside) xxx.yyy.zzz.10 192.168.128.100 netmask 255.255.255.255 0 2000			
static (dmz,outside) xxx.yyy.zzz.11 192.168.128.101 netmask 255.255.255.255 0 2000			
access-list acl_ouside deny tcp 10.10.10.0 255.255.255.0 host xxx.yyy.zzz.10 eq 80			
access-list acl_ouside deny tcp 10.10.10.0 255.255.255.0 host xxx.yyy.zzz.11 eq 80			
access-list acl_ouside deny tcp 10.10.20.0 255.255.255.0 host xxx.yyy.zzz.10 eq 80			
access-list acl_ouside deny tcp 10.10.20.0 255.255.255.0 host xxx.yyy.zzz.11 eq 80			
access-list acl_ouside deny tcp 192.168.128.0 255.255.255.0 host xxx.yyy.zzz.10 eq 80			
access-list acl_ouside deny tcp 192.168.64.0 255.255.255.0 host xxx.yyy.zzz.11 eq 80			
access-list acl_ouside permit tcp any gt 1023 host xxx.yyy.zzz.10 eq 80			
access-list acl_ouside permit tcp any gt 1023 host xxx.yyy.zzz.11 eq 80			

Configuration file

After developing these rules we need to organize them in the right order, because the Firewall reads the ACLs from top to down. This also requires to start with the most specific rule in the beginning . Also we need to eliminate some doubled rules in the NAT Translations from the lists.
After doing these steps we can apply the ACLs to the interface using the

access-group acl_outside in interface outside
access-group acl_inside in interface inside
access-group acl_dmz1 in interface dmz1
access-group acl_dmz2 in interface dmz2

commands and here is the complete firewall configuration:

enable password *Youre favourite password phrase, for configuration, enhanced troubleshooting*
passwd *youre 2nd favourite password phrase for basic admin and basic troubleshooting*
hostname *don´t call it pix version xx*
nameif ethernet0 outside security0
nameif ethernet1 inside security100
nameif ethernet2 *dmz1 security20*
nameif ethernet3 *dmz2 security40*

interface ethernet0 auto
interface ethernet1 auto
interface ethernet2 auto
interface ethernet3 auto

ip address outside xxx.yyy.zzz.1 255.255.255.224
ip address inside 10.10.10.1 255.255.255.0
ip address dmz1 192.168.128.1 255.255.255.0
ip address dmz2 192.168.64.1.255.255.255

route outside 0.0.0.0 0.0.0.0 xxx.yyy.zzz.2
route inside 10.10.20.0 255.255.255.0 10.10.10.2

Nat 0 10.10.10.100 255.255.255.255
Nat 0 10.10.10.128 255.255.255.128
Nat 0 10.10.20.0 255.255.255.0
Nat 0 192.168.128.200 255.255.255.255
Global 1 xxx.yyy.zzz.9
Nat 1 10.10.10.100 255.255.255.255
Nat 10 10.10.10.128 255.255.255.128
Global 2 xxx.yyy.zzz.12
Nat 2 192.168.64.200 255.255.255.255
static (inside,dmz1) 10.10.10.101 10.10.10.101 netmask 255.255.255.255 0 2000
static (inside,dmz1) 10.10.10.102 10.10.10.102 netmask 255.255.255.255 0 2000
static (dmz1,outside) xxx.yyy.zzz.13 192.168.128.200 netmask 255.255.255.255 0 2000
static (dmz,outside) xxx.yyy.zzz.10 192.168.128.100 netmask 255.255.255.255 0 2000
static (dmz,outside) xxx.yyy.zzz.11 192.168.128.101 netmask 255.255.255.255 0 2000

access-list acl_inside permit udp host 10.10.10.100 gt 1023 host 192.168.64.100 eq domain
access-list acl_inside permit udp 10.10.10.128 255.255.255.128 gt 1023 host 192.168.64.100 eq domain
access-list acl_inside permit udp host 10.10.10.100 gt 1023 host 194.25.0.60 eq domain
access-list acl_inside permit udp 10.10.10.128 255.255.255.128 gt 1023 host 194.25.0.60 eq domain
access-list acl_ouside permit tcp host 194.25.0.60 gt 1023 host xxx.yyy.zzz14 eq domain
access-list acl_inside permit tcp host 10.10.10.101 gt 1023 host 192.168.128.200 eq 25
access-list acl_inside permit tcp host 10.10.10.102 gt 1023 host 192.168.128.200 eq 25

access-list acl_inside permit tcp 10.10.10.128 255.255.255.128 gt 1023 host 192.168.128.101 eq 80
access-list acl_inside permit tcp 10.10.10.128 255.255.255.128 gt 1023 host 192.168.128.102 eq 80
access-list acl_inside permit tcp 10.10.10.128 255.255.255.128 gt 1023 host 192.168.128.101 eq 443
access-list acl_inside permit tcp 10.10.10.128 255.255.255.128 gt 1023 host 192.168.128.102 eq 443
access-list acl_inside permit tcp 10.10.20.0 255.255.255.0 gt 1023 host 192.168.128.101 eq 80
access-list acl_inside permit tcp 10.10.20.0 255.255.255.0 gt 1023 host 192.168.128.102 eq 443

Access-list acl_inside deny tcp 10.10.10.128 255.255.255.128 192.168.128.0 255.255.255.0 eq 80
Access-list acl_inside deny tcp 10.10.10.128 255.255.255.128 192.168.128.0 255.255.255.0 eq 443
Access-list acl_inside deny tcp 10.10.10.128 255.255.255.128 192.168.128.0 255.255.255.0 eq 21
Access-list acl_inside deny tcp 10.10.10.128 255.255.255.128 192.168.64.0 255.255.255.0 eq 80
Access-list acl_inside deny tcp 10.10.10.128 255.255.255.128 192.168.64.0 255.255.255.0 eq 443
Access-list acl_inside deny tcp 10.10.10.128 255.255.255.128 192.168.64.0 255.255.255.0 eq 21
access-list acl_inside permit tcp 10.10.10.128 255.255.255.128 gt 1023 any eq 80
access-list acl_inside permit tcp 10.10.10.128 255.255.255.128 gt 1023 any eq 443
access-list acl_inside permit tcp 10.10.10.128 255.255.255.128 gt 1023 any eq 21

access-list acl_dmz1 permit tcp host 192.168.128.200 gt 1023 host 10.10.10.101 eq 25
access-list acl_dmz1 permit tcp host 192.168.128.200 gt 1023 host 10.10.10.102 eq 25
access-list acl_dmz1 deny tcp host 192.168.128.200 any 10.10.10.0 255.255.255.0 eq 25
access-list acl_dmz1 deny tcp host 192.168.128.200 any 10.10.20.0 255.255.255.0 eq 25
access-list acl_dmz1 deny tcp host 192.168.128.200 any 192.168.64.0 255.255.255.0 eq 25

access-list acl_dmz1 permit tcp host 192.168.128.200 gt 1023 any eq 25
access-list acl_dmz1 permit udp host 192.168.128.200 gt 1023 host 192.168.64.100 eq domain

```
access-list acl_dmz2 permit udp host 192.168.64.200 gt 1023 host 194.25.0.60 eq domain
access-list acl_dmz2 permit udp host 192.168.64.100 gt 1023 host 194.25.0.60 eq domain
Access-list acl_dmz2 deny tcp host 192.168.64.200  192.168.128.0 255.255.255.0 eq 80
Access-list acl_dmz2 deny tcp host 192.168.64.200  192.168.128.0 255.255.255.0 eq 443
Access-list acl_dmz2 deny tcp host 192.168.64.200  10.10.10.0 255.255.255.0 eq 80
Access-list acl_dmz2 deny tcp host 192.168.64.200  10.10.10.0 255.255.255.0 eq 443
Access-list acl_dmz2 deny tcp host 192.168.64.200  10.10.20.0 255.255.255.0 eq 80
Access-list acl_dmz2 deny tcp host 192.168.64.200  10.10.20.0 255.255.255.0 eq 443
access-list acl_dmz2 permit tcp host 192.168.64.200 gt 1023 any eq 80
access-list acl_dmz2 permit tcp host 192.168.64.200 gt 1023 any eq 443
```

```
access-list acl_ouside deny  tcp 10.10.10.0 255.255.255.0 host xxx.yyy.zzz.10 eq 25
access-list acl_ouside deny  tcp 10.10.20.0 255.255.255.0 host xxx.yyy.zzz.10 eq 25
access-list acl_ouside deny  tcp 192.168.128.0  255.255.255.0 host xxx.yyy.zzz.10 eq 25
access-list acl_ouside deny  tcp 192.168.64.0  255.255.255.0 host xxx.yyy.zzz.11 eq 25
access-list acl_ouside deny  tcp 10.10.10.0 255.255.255.0 host xxx.yyy.zzz.10 eq 80
access-list acl_ouside deny  tcp 10.10.10.0 255.255.255.0 host xxx.yyy.zzz.11 eq 80
access-list acl_ouside deny  tcp 10.10.20.0 255.255.255.0 host xxx.yyy.zzz.10 eq 80
access-list acl_ouside deny  tcp 10.10.20.0 255.255.255.0 host xxx.yyy.zzz.11 eq 80
access-list acl_ouside deny  tcp 192.168.128.0  255.255.255.0 host xxx.yyy.zzz.10 eq 80
access-list acl_ouside deny  tcp 192.168.64.0  255.255.255.0 host xxx.yyy.zzz.11 eq 80
access-list acl_outside permit tcp any gt 1023 host xxx.yyy.zzz.13 eq 25
access-list acl_ouside permit tcp any gt 1023 host xxx.yyy.zzz.10 eq 80
access-list acl_ouside permit tcp any gt 1023 host xxx.yyy.zzz.11 eq 80
```

```
access-group acl_outside in interface outside
access-group acl_inside in interface inside
access-group acl_dmz1 in interface dmz1
access-group acl_dmz2 in interface dmz2
```

So do you think this is a lot? Just remember our network set up, this is only the configuration of a very simple example. Maybe it is helpful to have the rules together with the intended policy as in the lists above when the policy gets really complicated.

Further Steps

- Antispoofing
- Management
- Logging
- Test

An anti spoofing policy is already integrated in this configuration, because of the parental policy which says everything which is not specifically allowed is forbidden. So in our configuration are only NAT rules and access-lists for specific known networks and machines configured.
This will not prevent the clients or server from spoofed packets from the internet, but there will be no packets from the internet forwarded with a source address from our corporate network or our perimeter zone.

The network based management, the logging and the testing of the firewall are there

own pieces of the jigsaw and I only want to remember not to forget these steps.

Conclusion

Even if the firewall management tools get smarter it is helpful to have a basic understanding what the firewall actually is doing and to have a detailed procedure how to implement the policy in the firewall configuration.

With the documentation of all these steps its possible to see which command is driven by which policy, so if there are any problems or changes in the policy it will be easy to implement changes in the firewall configuration and also to update the documentation.

References

[1] Cisco Systems “Itrusion detection Planning Guide: Design Considerations”
http://www.cisco.com/univercd/cc/td/doc/product/iaabu/idpg/design.htm

[2] Cisco Systems “White Paper: Cisco's PIX Firewall and Stateful Firewall Security”
http://www.cisco.com/warp/public/cc/pd/fw/sqfw500/tech/nat_wp.htm

[3] Lance Spitzner “How Stateful is Stateful Inspection?”
http://www.enteract.com/~lspitz/fwtable.html

[4] IETF “RFC 1918”
http://www.ietf.org/rfc/rfc1918.txt?number=1918

[5] IETF “RFC 3022”
http://www.faqs.org/rfcs/rfc3022.html

[6] IETF “RFC 1700”
http://www.ietf.org/rfc/rfc1700.txt

[7] Robert J. Kohlhepp “Buildin a DNS”
http://www.nwc.com/netdesign/cook5.html

[8] DNSRD “Documents about DNS”
http://www.dns.net/dnsrd/docs/

[9] Miminun “Glossary”
http://secondary.com/doc/english//glossary-t.html#zonetransfer

[10]
E.D. Zwicky, S. Cooper, D.B. Chapman “Building Internet Firewalls”
O´Reilly , www.oreilly.com

Last Updated: July 23rd, 2017

Upcoming Training

SANS Prague 2017	Prague, Czech Republic	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Boston 2017	Boston, MA	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Salt Lake City 2017	Salt Lake City, UT	Aug 14, 2017 - Aug 19, 2017	Live Event
Community SANS Omaha SEC401*	Omaha, NE	Aug 14, 2017 - Aug 19, 2017	Community SANS
SANS New York City 2017	New York City, NY	Aug 14, 2017 - Aug 19, 2017	Live Event
SANS Chicago 2017	Chicago, IL	Aug 21, 2017 - Aug 26, 2017	Live Event
SANS Virginia Beach 2017	Virginia Beach, VA	Aug 21, 2017 - Sep 01, 2017	Live Event
SANS Adelaide 2017	Adelaide, Australia	Aug 21, 2017 - Aug 26, 2017	Live Event
Community SANS Trenton SEC401	Trenton, NJ	Aug 21, 2017 - Aug 26, 2017	Community SANS
Virginia Beach 2017 - SEC401: Security Essentials Bootcamp Style	Virginia Beach, VA	Aug 21, 2017 - Aug 26, 2017	vLive
Community SANS Pasadena SEC401 @ NASA	Pasadena, CA	Aug 23, 2017 - Aug 30, 2017	Community SANS
Mentor Session - SEC401	Minneapolis, MN	Aug 29, 2017 - Oct 10, 2017	Mentor
SANS San Francisco Fall 2017	San Francisco, CA	Sep 05, 2017 - Sep 10, 2017	Live Event
SANS Tampa - Clearwater 2017	Clearwater, FL	Sep 05, 2017 - Sep 10, 2017	Live Event
Mentor Session - SEC401	Edmonton, AB	Sep 06, 2017 - Oct 18, 2017	Mentor
SANS Network Security 2017	Las Vegas, NV	Sep 10, 2017 - Sep 17, 2017	Live Event
Community SANS Albany SEC401	Albany, NY	Sep 11, 2017 - Sep 16, 2017	Community SANS
Mentor Session - SEC401	Ventura, CA	Sep 11, 2017 - Oct 12, 2017	Mentor
Community SANS Columbia SEC401	Columbia, MD	Sep 18, 2017 - Sep 23, 2017	Community SANS
Community SANS Dallas SEC401	Dallas, TX	Sep 18, 2017 - Sep 23, 2017	Community SANS
Community SANS Boise SEC401	Boise, ID	Sep 25, 2017 - Sep 30, 2017	Community SANS
Baltimore Fall 2017 - SEC401: Security Essentials Bootcamp Style	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	vLive
Community SANS New York SEC401	New York, NY	Sep 25, 2017 - Sep 30, 2017	Community SANS
Rocky Mountain Fall 2017	Denver, CO	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS London September 2017	London, United Kingdom	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS Baltimore Fall 2017	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS Copenhagen 2017	Copenhagen, Denmark	Sep 25, 2017 - Sep 30, 2017	Live Event
Community SANS Sacramento SEC401	Sacramento, CA	Oct 02, 2017 - Oct 07, 2017	Community SANS
SANS DFIR Prague 2017	Prague, Czech Republic	Oct 02, 2017 - Oct 08, 2017	Live Event
Community SANS Charleston SEC401	Charleston, SC	Oct 02, 2017 - Oct 07, 2017	Community SANS
Mentor Session - SEC401	Arlington, VA	Oct 04, 2017 - Nov 15, 2017	Mentor