思科 ASA 和 PIX 防火墙配置手册

根据互联网资料整理
By Mast
2007 年 11 月 9 日

目　　录

第一章　配置基础

1.1 用户接口

思科防火墙支持下列用户配置方式：

Console，Telnet，SSH（1.x 或者 2.0，2.0 为 7.x 新特性，PDM 的 http 方式（7.x 以后称为 ASDM）和 VMS 的 Firewall Management Center。

支持进入 Rom Monitor 模式，权限分为用户模式和特权模式，支持 Help，History 和命令输出的搜索和过滤。

注：Catalyst6500 的 FWSM（防火墙服务模块 Firewall Service Module）没有物理接口接入，通过下面 CLI 命令进入：

Switch# session slot slot processor 1 （FWSM 所在 slot 号）

用户模式：

Firewall> 为用户模式，输入 enable 进入特权模式 Firewall#。特权模式下可以进入配置模式，在 6.x 所有的配置都在一个全局模式下进行，7.x 以后改成和 IOS 类似的全局配置模式和相应的子模式。通过 exit，ctrl-z 退回上级模式。

配置特性：

在原有命令前加 no 可以取消该命令。Show running-config 或者 write terminal 显示当前配置，7.x 后可以对 show run 的命令输出进行搜索和过滤。Show running-config all 显示所有配置，包含缺省配置。Tab 可以用于命令补全，ctrl-l 可以用于重新显示输入的命令（适用于还没有输入完命令被系统输出打乱的情况），help 和 history 相同于 IOS 命令集。

Show 命令支持 begin，include，exclude，grep 加正则表达式的方式对输出进行过滤和搜索。

Terminal width 命令用于修改终端屏幕显示宽度，缺省为 80 个字符，pager 命令用于修改终端显示屏幕显示行数，缺省为 24 行，pager lines 0

命令什么效果可以自己试试。

1.2 防火墙许可介绍

防火墙具有下列几种许可形式，通过使用 show version 命令可以看设备所支持的特性：

Unrestricted (UR) 所有的限制仅限于设备自身的性能，也支持 Failover

Restricted (R) 防火墙的内存和允许使用的最多端口数有限制，不支持 Failover

Failover (FO) 不能单独使用的防火墙，只能用于 Failover

Failover-Active/Active (FO-AA) 只能和 UR 类型的防火墙一起使用，支持 active/active failover

注：FWSM 内置 UR 许可。

activation-key 命令用于升级设备的许可，该许可和设备的 serial number 有关（show version 输出可以看到），6.x 为 16 字节，7.x 为 20 字节。

1.3 初始配置

跟路由器一样可以使用 setup 进行对话式的基本配置。

第二章　配置连接性

2.1 配置接口

接口基础：

防火墙的接口都必须配置接口名称，接口 IP 地址和掩码（7.x 开始支持 IPv6）和安全等级。接口可以是物理接口也可以是逻辑接口（vlan），从 6.3 开始支持 SPAN、trunk，但只支持 802.1Q 封装，不支持 DTP 协商。

接口基本配置：

注：对于 FWSM 所有的接口都为逻辑接口，名字也是 vlan 后面加上 vlanid。例如 FWSM 位于 6500 的第三槽，配置三个接口，分别属于 vlan 100,200,300.

Switch(config)# firewall vlan-group 1 100,200,300

Switch(config)# firewall module 3 vlan-group 1

Switch(config)# exit

Switch# session slot 3 processor 1

经过此配置后形成三个端口 vlan100,vlan200,vlan300

PIX 6.x：

Firewall(config)# interface hardware-id [hardware-speed] [shutdown] （Hardware-id 可以用 show version 命令看到）

PIX 7.x：

Firewall(config)# interface hardware-id

Firewall(config-if)# speed {auto | 10 | 100 | nonegotiate}

Firewall(config-if)# duplex {auto | full | half}

Firewall(config-if)# [no] shutdown

命名接口：

FWSM 2.x：

Firewall(config)# nameif vlan-id if_name securitylevel

PIX 6.x：

Firewall(config)# nameif {hardware-id | vlan-id} if_name securitylevel

PIX 7.x：

Firewall(config)# interface hardware_id[.subinterface]

Firewall(config-if)# nameif if_name

Firewall(config-if)# security-level level

注：Pix 7.x 和 FWSM 2.x 开始支持不同接口有相同的 security level，前提是全局配置模式下使用 same-security-traffic permit inter-interface 命令。

配置 IP 地址：

静态地址：

Firewall(config)# ip address if_name ip_address [netmask]

动态地址：

Firewall(config)# ip address outside dhcp [setroute] [retry retry_cnt]

注：setroute 参数可以同时获得来自 DHCP 服务器的缺省路由，再次输入此命令可以 renew 地址。

PPPOE：

Firewall(config)# vpdn username JohnDoe password JDsecret

Firewall(config)# vpdn group ISP1 localname JohnDoe

Firewall(config)# vpdn group ISP1 ppp authentication chap

Firewall(config)# vpdn group ISP1 request dialout pppoe

Firewall(config)# ip address outside pppoe setroute

验证接口：

Firewall# show ip

IPv6 地址配置（7.x 新特性）

暂略

ARP 配置

配置一个静态的 ARP 条目：

Firewall(config)# arp if_name ip_address mac_address [alias]

配置 timeout 时间：

Firewall(config)# arp timeout seconds 缺省为 4 小时

注：一般情况下使用 clear arp 会清除所有的 ARP 缓存，不能针对单个的条目，但是可以通过以下变通方法：配置一个静态的条目，映射有问题的 ip 为一个假的 mac 地址，然后 no 掉该命令就会重新建立一个 arp 条目。

MTU 和分段

配置 MTU：

Firewall(config)# mtu if_name bytes

使用 show mtu (6.3) 或者 show running-config mtu (7.x)来验证分段（fragment）的几个命令：

限制等待重组的分段数：

Firewall(config)# fragment size database-limit [if_name]

限制每个包的分段数：

Firewall(config)# fragment chain chain-limit [if_name]

限制一个数据包分段到达的时间：

Firewall(config)# fragment timeout seconds [if_name]

配置接口的优先队列（7.x 新特性）

暂略

2.2 配置路由

启用 PRF 防止地址欺骗：

Firewall(config)# ip verify reverse-path interface if_name

配置静态路由：

Firewall(config)# route if_name ip_address netmask gateway_ip [metric]

配置 RIP 被动听 RIP 更新(v1，v2)：

Firewall(config)# rip if_name passive [version 1] （Firewall(config)# rip if_name

passive version 2 [authentication [text | md5 key (key_id)]]）

宣告该接口为缺省路由：

Firewall(config)# rip if_name default version [1 | 2 [authentication [text | md5 key key_id]]

配置 OSPF：

定义 OSPF 进程：

Firewall(config)# router ospf pid

指定相应网络到 OSPF 区域：

Firewall(config-router)# network ip_address netmask area area_id

可选：定义 Router ID：

Firewall(config-router)# router-id ip_address

记录 OSPF 邻居状态更新：

Firewall(config-router)# log-adj-changes [detail]

启用 OSPF 更新认证：

Firewall(config-router)# area area_id authentication [message-digest]

宣告缺省路由：

Firewall(config-router)# default-information originate [always] [metric value] [metric-type {1 | 2}] [route-map name]

调节 OSPF 参数：

Firewall(config-router)# timers {spf spf_delay spf_holdtime |lsa-group-pacing seconds}

2.3 DHCP

配置成为 DHCP Server：

配置地址池：

Firewall(config)# dhcpd address ip1[-ip2] if_name （最多 256 个客户端）

配置 DHCP 参数：

Firewall(config)# dhcpd dns dns1 [dns2]

Firewall(config)# dhcpd wins wins1 [wins2]

Firewall(config)# dhcpd domain domain_name

Firewall(config)# dhcpd lease lease_length Firewall(config)# dhcpd ping_timeout timeout

启用 DHCP 服务：

Firewall(config)# dhcpd enable if_name

验证：show dhcdp, show dhcpd bindings, show dhcpd statistics

配置 DHCP 中继：

定义真实 DHCP Server：

Firewall(config)# dhcprelay server dhcp_server_ip server_ifc(最多 4 个)

中继参数：

Firewall(config)# dhcprelay timeout seconds

Firewall(config)# dhcprelay setroute client_ifc

启用中继：

Firewall(config)# dhcprelay enable client_ifc

验证 show dhcprelay statistics

2.4 组播的支持

暂略

第三章　防火墙的管理

3.1 使用 Security Context 建立虚拟防火墙（7.x 特性）

特性介绍：从 PIX7.0 和 FWSM 2.2(1)开始，可以把物理的一个防火墙配置出多个虚拟的防火墙，每个防火墙称为 context，这样一个防火墙就支持两种工作模式：single-context 和 multiple-context，处于后者工作模式的防火墙被分为三个功能模块：system execution space(虽然没有 context 的功能，但是是所有的基础)，administrative context(被用来管理物理的防火墙) 和 user contexts(虚拟出来的防火墙，所有配置防火墙的命令都适用)

配置：首先使用 show activation-key 来验证是否有 multiple-context 的许可，然后通过 mode multiple 和 mode single 命令在这两个模式之间进行切换，当然也可以用 show mode 来验证现在工作在什么模式下。在不同 context 下进行切换使用：

Firewall# changeto {system | context name}

由于所有的 context 的定义都必须在 system execution space 下，所以要首先使用 changeto system 转入该模式：

Firewall(config)#context name

接着要把物理接口映射到 context 中 只要这样才能在相应的 context 下显示出物理接口，从而配置其属性：

Firewall(config-ctx)# allocate-interface physical-interface [map-name]

最后定义 context 的 startup-config 的存放位置：

Firewall(config-ctx)# config-url url

通过 show context 验证

注：当防火墙工作在 multiple-context 模式下，admin context 就自动生成。(show context 来验证)由于所有的 context 都共享设备的资源，所以要限制各个 context 的资源分配首先定义 class：

Firewall(config)# class name

然后

Firewall(config-class)# limit-resource all number%

Firewall(config-class)#limit-resource [rate] resource_name number[%]

最后在相应的 context 配置下

Firewall(config-ctx)# member class

通过以下命令验证 show class， show resource allocation, show resource usage 等

注：缺省 telnet，ssh，IPsec 5 sessions，MAC address 65535 条目

3.2 管理 Flash 文件系统

6.x 文件系统只有六种文件可以保存到 Flash，没有文件名只有代号，没有目录结构：

0-OS 镜像 1-启动文件 2-VPN 和密匙证书 3-PDM 镜像 4-崩溃信息 5-0 的文件大小

show flashfs 显示 flash 文件；

7.x 和 FWSM 文件系统：

7.x 和 FWSM 更像 IOS 的文件系统，具有层级目录，要被格式化后才可以使用，7.x 使用 flash:/代表 Flash 文件系统，FWSM 分别使用 flash:/ (系统镜像)和 disk:/(配置文件)。由于该系统使用类 Unix 的指令，所以可以使用下列常用命令来对该文件系统操作：

dir pwd cd more delete copy rename mkdir rmdir format erase fsck(检查文件系统完整性)。

6.x 在 Flash 裡面只能保存一个系统镜像，7.x 则废除了此种限制通过使用

Firewall(config)# boot system flash:filename

来选取不同的系统镜像，show bootvar 进行验证

OS 升级 见附录

3.3 管理配置文件

7.0 以后可以使用多个启动配置文件：

Firewall(config)# boot config url

显示启动配置文件：

Firewall# show startup-config

Firewall# show configuration (6.x 为 show configure)

保存当前配置文件 write memory, copy running-config startup-config, write net [[server-ip-address]:[filename]] (7.x 也支持 copy 至 tftp)

强制 standby 同步当前配置文件 write standby 删除启动配置文件 write erase

合并启动配置文件为当前配置文件 configure memory 从 Web 导入配置文件 configure http[s]://[user:password@]location[:port]/ http-pathname (7.x 支持 copy 自以上源)

合并配置文件自自动更新服务器：

Firewall(config)# auto-update device-id {hardware-serial | hostname | ipaddress [if_name] | mac-address [if_name] | string text}

Firewall(config)# auto-update server http[s]://[username:password@]AUSserver-IP-address[:port]/autoupdate/AutoUpdateServlet[verify-certificate]

3.4 管理管理会话

Firewall(config)# console timeout minutes 配置 console 登录的超时(缺省 0 不超时)

禁止来自 outside 端口的 telnet，启用 telnet：

Firewall(config)# telnet ip_address netmask if_name

Firewall(config)# telnet timeout minutes 配置 telnet 超时

启用 SSH 配置

首先生成 RSA 密钥对：

Firewall(config)# domain-name name

Firewall(config)# ca generate rsa key [modulus] (7.x 使用 crypto key generate rsa general-keys [modulus modulus])

Firewall(config)# ca save all (7.x 自动保存)

使用 show ca mypubkey rsa 来验证(7.x show crypto key mypubkey rsa) ca zeroize rsa 作废原有密匙对(7.x crypto key zeroize rsa default)

最后允许 ssh 会话：

Firewall(config)# ssh ip_address netmask if_name

ssh version 命令可以选择 ssh 的版本，ssh timeout 定义超时时间；

PDM/ASDM 配置：

由于 PDM 存放位置固定，所以不需要指定镜像的位置，ASDM 使用：

Firewall(config)# asdm image device:/path

来指定镜像位置，如果没有可以使用 copy 命令来安装。然后配置访问许可：

Firewall# http ip_address subnet_mask if_name

启用 HTTP 进程：

Firewall# http server enable

使用 https://ip-address/admin 来访问。

Banner 配置：

Firewall(config)# banner {exec | login | motd} text　对 banner 不能修改，只能用 no 来删除，或者 clear banner 来清除所有的 banner（7.0 clear configure banner）

监控管理会话：　who 监控 telnet 会话　kill telnet-id 来清除会话，show ssh sessions 监控 ssh 会话，ssh disconnect session-id 清除 ssh 会话，show pdm sessions 监控 pdm 会话，pdm disconnect session-id 清除 pdm 会话

3.5 系统重启和崩溃

通常使用 reload 命令重启系统，从 7.0 以后支持在特定的时间重启系统：

Firewall# reload at hh:mm [month day | day month] [max-hold-time {minutes |

hhh:mm}] [noconfirm] [quick] [save-config] [reason text]

或者经过一定的时间间隔后重启：

Firewall# reload in {minutes | hh:mm} [max-hold-time {minutes | hhh:mm}] [noconfirm] [quick] [save-config] [reason text]

启用崩溃信息生成：

Firewall(config)# crashinfo save enable (7.0 no crashinfo save disable)

show crashinfo 来看崩溃信息

clear crashinfo 删除信息（FWSM 使用 crashdump）

3.6 SNMP 支持

系统 SNMP 信息 ：

Firewall(config)# snmp-server location string (contact string)

SNMP 访问许可 ：

Firewall(config)# snmp-server host if_name ip_addr [poll | trap]

Firewall(config)# snmp-server community key

第四章　用户管理

4.1 一般用户管理

注：缺省情况下认证用户仅需要 password，这样的一般用户缺省用户名就是 enable_1,在 ssh 情况下缺省用户名就是 pix，然后用 password 来认证。

非特权模式密码配置：

Firewall(config)# {password | passwd} password [encrypted] (恢复缺省密码 cisco 用 clear {password | passwd})

特权模式密码配置：

Firewall(config)# enable password [pw] [level priv_level] [encrypted]

4.2 本地数据库管理用户

定义用户：

Firewall(config)# username username [{nopassword | password password}[encrypted]] privilege level

启用本地认证：

Firewall(config)# aaa authentication {serial | telnet | ssh | http} console LOCAL

注：缺省情况特权模式密码使用 enable password 定义，这样用户通过认证后使用 enable 来进入特权模式，而不管用户初始什么等级的权限，所有用户使用相同的密码。这里也可以使用本地 enable 认证(aaa authentication enable console LOCAL)，用户使用 username password 的密码来进入 enable，用户 enable 密码独立从而增加安全性。

本地授权：

Firewall(config)# aaa authorization command LOCAL

配置命令的特权等级：

Firewall(config)# privilege {show | clear | configure} level level [mode {enable | configure}] command command

使用 show privilege 来看当前命令的特权等级(7.x 使用 show run all privilege)

4.3 使用 AAA 服务器来管理用户

定义 AAA 服务器组和协议：

Firewall(config)# aaa-server server_tag protocol {tacacs+ | radius} (7.x 还增加了 kerberos,ldap,nt,sdi 协议的支持)

加入服务器到组：

Firewall(config)# aaa-server server_tag [(if_name)] host server_ip [key] [timeout seconds]

可选命令：

定义服务器失败阀值：

FWSM：Firewall(config)# aaa-server server_tag max-attempts number

PIX 6.x：Firewall(config)# aaa-server server_tag max-failed-attempts number

PIX 7.x：Firewall(config-aaa-server-group)# max-failed-attempts number

定义统计策略(7.x 特性)：

Firewall(config-aaa-server-group)# accounting-mode {single | simultaneous}

具体各协议参数配置暂略

4.4 配置 AAA 管理用户

启用认证：

Firewall(config)# aaa authentication {serial | telnet | ssh | http} consoleserver_tag [LOCAL]

启用授权：

Firewall(config)# aaa authorization command server_tag [LOCAL]

启用统计：

Firewall(config)# aaa accounting command [privilege level] server_tag

注：AAA 服务器配置略

4.5 配置 AAA 支持用户 Cut-Through 代理

4.6 密码恢复

第五章　防火墙的访问控制

5.1 防火墙的透明模式

特性介绍：从 PIX 7.0 和 FWSM 2.2 开始防火墙可以支持透明的防火墙模式，接口不需要配置地址信息，工作在二层。只支持两个接口 inside 和 outside，当然可以配置一个管理接口，但是管理接口不能用于处理用户流量，在多 context 模式下不能复用物理端口。由于连接的是同一地址段的网络，所以不支持 NAT，虽然没有 IP 地址但是同样可以配置 ACL 来检查流量。进入透明模式 Firewall(config)# firewall transparent (show firewall 来验证当前的工作模式，由于路由模式和透明模式工作方式不同，所以互相切换的时候会清除当前配置文件)。

配置接口：

Firewall(config)# interface hardware-id

Firewall(config-if)# speed {auto | 10 | 100 |nonegotiate}

Firewall(config-if)# duplex {auto | full | half}

Firewall(config-if)# [no] shutdown

Firewall(config-if)# nameif if_name

Firewall(config-if)# security-level level

注：不用配置 IP 地址信息，但是其它的属性还是要配置的，接口的安全等级一般要不一样，same-security-traffic permit inter-interface 命令可以免除此限制。

配置管理地址：

Firewall(config)# ip address ip_address subnet_mask

Firewall(config)# route if_name foreign_network foreign_mask gateway [metric]

MAC 地址表的配置：

Firewall# show mac-address-table 显示 MAC 地址表

Firewall(config)# mac-address-table aging-time minutes 设置 MAC 地址表过期时间

Firewall(config)# mac-address-table static if_name mac_address 设置静态 MAC 条目

Firewall(config)# mac-learn if_name disable 禁止特定接口地址学习(show mac-learn 验证)

ARP 检查：

Firewall(config)# arp if_name ip_address mac_address 静态 ARP 条目

Firewall(config)# arp-inspection if_name enable [flood | no-flood] 端口启用 ARP 检查

为非 IP 协议配置转发策略：

Firewall(config)# access-list acl_id ethertype {permit | deny} {any | bpdu | ipx | mpls-unicast | mpls-multicast | ethertype}

Firewall(config)# access-group acl_id {in | out} interface if_name

5.2 防火墙的路由模式和地址翻译

特性介绍：从高安全等级到低安全等级的访问称为 outbound 访问，需要配置地址翻译和 outbound 访问控制，PIX 缺省情况下不用配置 ACL 就允许此类访问，FWSM 则需要配置 ACL 来允许此类型的访问。而从低安全等级到高安全等级的访问称为 inboud 访问，也需要配置地址翻译和 inboud 访问控制，此类型必须配置 ACL.同一安全等级的访问也可以配置地址翻译。

支持下列几种 NAT 类型

Translation Type

Application

Basic Command

Direction in Which Connections Can Be Initiated

Static NAT

Real source addresses (and ports) are translated to mapped addresses (and ports) static

Inbound or outbound

Policy NAT

Conditionally translates real source addresses (and ports) to mapped addresses static

access-list

Inbound or outbound

Identity NAT

No translation of real source addresses nat 0

Outbound only

NAT exemption

No translation of real source addresses matched by the access list nat 0 access-list

Inbound or outbound

Dynamic NAT

Translates real source addresses to a pool of mapped addresses nat id global id address-range

Outbound only

PAT

Translates real source addresses to a single mapped address with dynamic port numbers nat id global id address

Outbound only

配置：

对于连接数的控制 PIX 6.x ... [norandomseq] [max_conns [emb_limit]]

PIX 7.x ... [norandomseq] [[tcp] max_conns [emb_limit]] [udp udp_max_conns]

连接超时控制 Firewall(config)# timeout [conn hh:mm:ss] [udp hh:mm:ss]

静态 NAT：

基于地址的静态翻译：

Firewall(config)# static (real_ifc,mapped_ifc) {mapped_ip | interface} {real_ip [netmask mask]} [dns] [norandomseq] [max_conns [emb_limit]]

基于端口的静态翻译：

Firewall(config)# static (real_ifc,mapped_ifc) {tcp | udp} {mapped_ip | interface} mapped_port {real_ip real_port [netmask mask]} [dns] [norandomseq] [max_conns

[emb_limit]]

策略 NAT：

定义翻译策略：

Firewall(config)# access-list acl_name permit ip real_ip real_mask foreign_ip foreign_mask

静态的：

Firewall(config)# static (real_ifc,mapped_ifc) mapped_ip access-list acl_name [dns] [norandomseq] [max_conns [emb_limit]]

NAT 的：

Firewall(config)# global (mapped_ifc) nat_id {global_ip [-global_ip] [netmask global_mask]} | interface

Firewall(config)# nat (real_ifc) nat_id access-list acl_name [dns] [outside][norandomseq] [max_conns [emb_limit]]

Identify NAT Firewall(config)# nat (real_ifc) 0 real_ip real_mask [dns] [norandomseq] [max_conns [emb_limit]]

注：nat 0 和 static 相同地址的区别在于：nat 0 只能用于 outbound 访问，static 两种访问都可以，对同一地址不建议同时配置此两类命令。

NAT Exemption

Firewall(config)# access-list acl_name permit ip local_ip local_mask foreign_ip foreign_mask

Firewall(config)# nat (real_ifc) 0 access-list acl_name [dns] [outside] [max_conns [emb_limit] [norandomseq]]

注：此类型 NAT 策略只能根据源和目的地址不能根据协议类型或者端口

动态地址翻译

定义 NAT 的映射地址：

Firewall(config)# global (mapped_ifc) nat_id global_ip[-global_ip] [netmask global_mask]

定义 PAT 的映射地址：

Firewall(config)# global (mapped_ifc) nat_id {global_ip | interface}

定义翻译策略：

Firewall(config)# nat (real_ifc) nat_id real_ip [mask [dns] [outside] [[norandomseq] [max_conns [emb_limit]]]

注：也可以使用 ACL 来做类似的策略 NAT。

5.3 使用 ACL 进行访问控制

特性介绍：防火墙的 ACL 配置跟 IOS 不同，子网掩码部分为正常的子网掩码不需要使用反转的子网掩码。还支持 Object group，包含 IP 地址组，

ICMP 类型组，IP 协议或者端口组，并且支持组嵌套。access-list acl_name compiled 配置 Turbo ACL，7.x 自动 turbo。防火墙的 ACL 缺省是扩展模式的，7.x 后也支持标准模式了尽管只用于路由协议的配置上，并且加上了 extend 的参数，虽然配置的时候可以不必强制用这个参数但是当你需要移除该条目的时候要记得把 extend 这个参数加上。

配置

定义 Object Group

网络对像组

Firewall(config)# object-group network group_id

Firewall(config-network)# description text

Firewall(config-network)# network-object ip_addr mask (或者 host ip_addr)

Firewall(config-network)# group-object group_id

ICMP 对像组

Firewall(config)# object-group icmp-type group_id

Firewall(config-icmp-type)# description text

Firewall(config-icmp-type)# icmp-object icmp_type

Firewall(config-icmp-type)# group-object group_id

协议对像组

Firewall(config)# object-group protocol group_id

Firewall(config-protocol)# description text

Firewall(config-protocol)# protocol-object protocol

Firewall(config-protocol)# group-object group_id

服务对像组

Firewall(config)# object-group service group_id {tcp | udp | tcp-udp}

Firewall(config-service)# description text

Firewall(config-service)# port-object range begin_port end_port (或者 eq port)

Firewall(config-service)# group-object group_id

定义时间范围 7.0 特性

Firewall(config)# time-range name

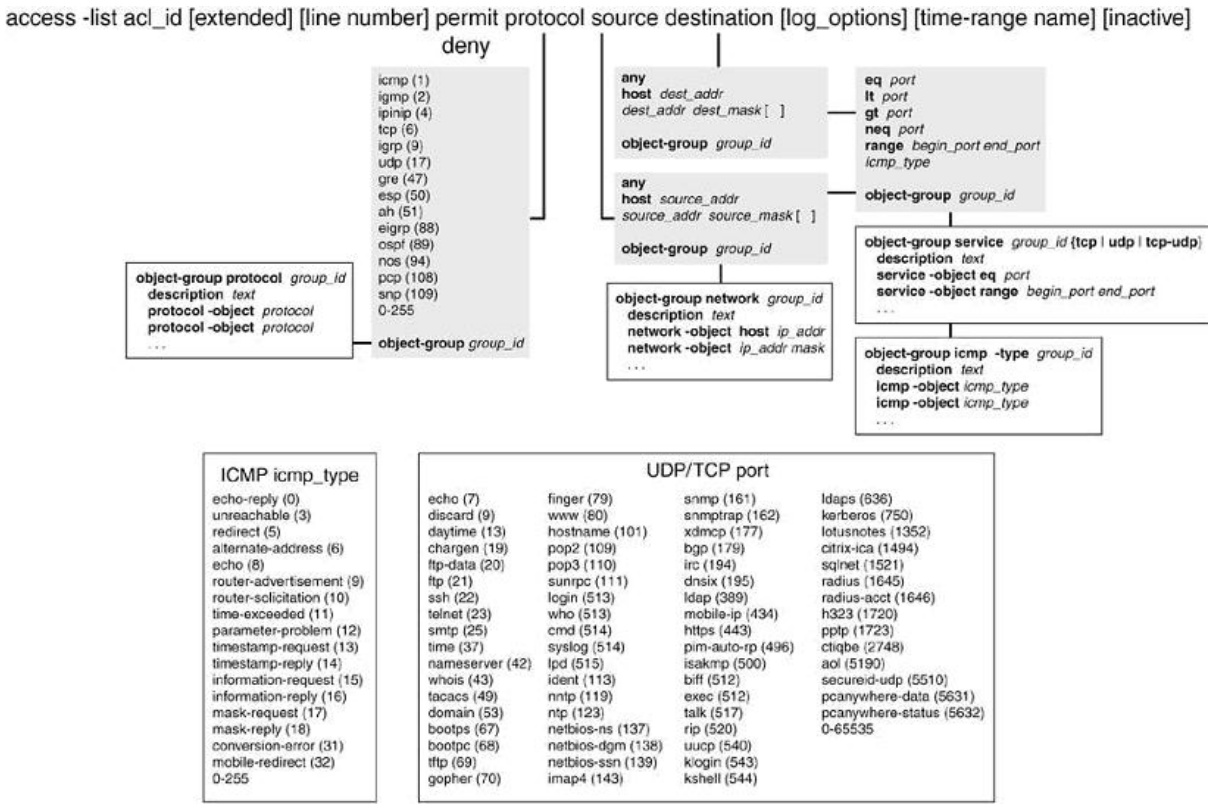
Firewall(config-time-range)# periodic start-day hh:mm to end-day hh:mm

Firewall(config-time-range)# periodic days-of-the-week hh:mm to hh:mm

Firewall(config-time-range)# absolute [start hh:mm day month year] [end hh:mm day month year]

配置 ACL

Firewall(config)# access-list acl_id [line line-num] [extended] {permit | deny} {protocol | object-group protocol_obj_group} {source_addr source_mask | object-group network_obj_group} [operator sport | object-group service_obj_group] {destination_addr destination_mask |object-group network_obj_group} [operator dport | object-group service_obj_group] [log [[disable | default] | [level]]] [interval secs]] [time-range name] [inactive]



show access-list 来验证， clear access-list acl_id counters 重置 ACL 计数器

第六章　配置 Failover 增加可用性

特性介绍：为了增强可用性，避免单点故障，提高性能等原因才引入了 Failover 的特性。Active-Standby 是最初支持的一种特性，其中一台是 UR 的许可，另一台为 UR 或者 Failover-only 的许可，FWSM 缺省支持此种模式，在该模式下一个为 Active 工作状态，Standby 只是监控 Active 的状态而不工作，这样就在性能上虽然有两台设备但是并没有得到加强。在 7.x 以后由于引入了 context 的概念，这样 Active-Active 另一种 Failover 的特性也出现了，在每个 context 下有自己的 active 和 standby，配置每个设备在不同 context 下的角色从而使其都工作，也增加了性能，但是此模式只被 PIX515E，525,535 和 ASA 平台支持。

6.1 配置 Failover

确定主备用的设备：一种方式是通过不同的许可来决定，如果两者都是 UR 的许可，对于适用 serial 连接的根据线缆两端的主备用标识来决定，如果适用 lan 的话使用下面的命令来决定：

Firewall(config)# failover lan unit {primary | secondary}

配置 lan 使用的端口：

FWSM 2.x：

Firewall(config)# failover interface ip if_name ip_address mask standby ip_address

PIX 6.x：

Firewall(config)# interface phy_if phy_speed

Firewall(config)# nameif phy_if if_name securitylevel

Firewall(config)# ip address if_name ip_address netmask

Firewall(config)# failover ip address if_name ip_address

PIX 7.x：

Firewall(config)# interface phy_if

Firewall(config-if)# speed speed

Firewall(config-if)# duplex duplex

Firewall(config-if)# no shutdown

Firewall(config-if)# exit

Firewall(config)# failover interface ip if_name ip_address mask standby ip_address

定义用于 Failover 通讯的接口：

FWSM 2.x：

Firewall(config)# failover lan interface if_name vlan vlan

PIX 6.x：

Firewall(config)# failover lan interface if_name

PIX 7.x：

Firewall(config)# failover lan interface if_name phy_if

也可以使用 failover lan key key-string 命令对通讯进行加密

failover lan enable 启用 lan-based failover，FWSM 缺省使用此模式，不需要此命令。

对于 Active-Active 模式需要在主设备的 system execution space 下配置 Failover 组：

Firewall(config)# failover group {1 | 2}

Firewall(config-fover-group)# {primary | secondary}

Firewall(config-fover-group)# preempt

对接口使用虚拟的 MAC 地址：

PIX 6.x：

Firewall(config)# failover mac address if_name active_mac standby_mac

PIX 7.x(A-S)：

Firewall(config)# failover mac address phy_if active_mac standby_mac

PIX 7.x (A-A)：

Firewall(config)# failover group {1 | 2}

Firewall(config-fover-group)# mac address phy_if active_mac standby_mac

定义健康监控策略：

PIX 6.x：

Firewall(config)# failover poll time

PIX 7.x：

Firewall(config)# failover polltime [unit] [msec] time [holdtime holdtime]

Firewall(config)# failover polltime interface time

Firewall(config)# failover interface-policy num[%]

Firewall(config)# monitor-interface if_name

保持 HTTP 的状态信息：

Firewall(config)# failover replicate http

Firewall(config)# failover 启用 Failover 进程

6.2 管理 Failover

show failover 命令对状态进行监控，后面可以加 state,lan, history,等参数。

(no)Failover active 手动的对状态进行切换,重置一个失败的设备 failover reset。对于不能同步的挂起设备使用 failover reload-standby 强制重启。

6.3 升级 Failover 模式防火墙的 OS 镜像

见附录

第七章　配置负载均衡

特性介绍：虽然使用 Failover 保证了高可用性，但是在流量分担上还有劣势，尽管 7.x 支持了 A-A 的模式，不过也仅仅只能是两台防火墙。真正的负载均衡的实现有三种方式，第一为软件方式，使用 6500 平台上 IOS SLB(Server Load Balancing)特性的一个子集 FWLB 来实现，第二为硬件方式，在 6500 上配置 CSM(Content Switching Module)来实现，最后一种为专属设备方式，思科的 CSS(Content Services Switch)产品线的设备来实现。要注意的是在配置负载均衡的时候要 inside 和 outside 同时配置，免得出现来回链路不同而被丢弃的情况。

7.1 配置软件实现 (只在 6500 native ios 模式下)

定义防火墙群的连接性：

Router(config)# vlan vlan-id

Router(config)# interface vlan vlan-id

Router(config-if)# ip address ip-address subnet-mask

Router(config-if)# no shutdown

Router(config)# ip route inside-network subnet-mask fw-outside-address

定义针对每个防火墙失败的探测器：

Router(config)# ip slb probe name ping

Router(config-slb-probe)# address ip-address

Router(config-slb-probe)# interval seconds

Router(config-slb-probe)# faildetect retry-count

定义防火墙群：

Router(config)# ip slb firewallfarm firewallfarm-name

Router(config-slb-fw)# real ip-address

Router(config-slb-fw-real)# probe probe-name

Router(config-slb-fw-real)# inservice

Router(config-slb-fw-real)# weight weighting-value

定义特定的数据流到该防火墙群(只针对 Outside)：

Router(config-slb-fw)# access [source source-ip-address network-mask] [destination destination-ip-address network-mask]

选择 FWLB 的方式：

Router(config-slb-fw)# predictor hash address [port]

启用 FWLB：

Router(config-slb-fw)# inservice

7.2 配置硬件实现

进入 CSM 负载均衡模式：

Switch(config)# ip slb mode csm

选择 CSM 模块：

Switch(config)# module csm slot-number

配置到离开防火墙群的连接性：

Switch(config-module-csm)# vlan vlan-id client

Switch(config-slb-vlan-client)# ip address ip-address netmask

Switch(config-slb-vlan-client)# gateway ip-address

Switch(config-slb-vlan-client)# exit

配置到防火墙群的连接性：

Switch(config-module-csm)# vlan vlan-id server

Switch(config-slb-vlan-server)# ip address ip-address netmask

Switch(config-slb-vlan-server)# alias ip-address netmask

Switch(config-slb-vlan-server)# route ip-address netmask gateway gw-ip-address

定义防火墙探测器：

Switch(config-module-csm)# probe probe-name icmp

Switch(config-slb-probe-icmp)# interval seconds

Switch(config-slb-probe-icmp)# receive receive-timeout

Switch(config-slb-probe-icmp)# retries retry-count

Switch(config-slb-probe-icmp)# failed failed-interval

定义防火墙群：

Switch(config-module-csm)# serverfarm serverfarm-name

Switch(config-slb-sfarm)# real ip-address

Switch(config-slb-real)# inservice

Switch(config-slb-sfarm)# predictor hash address {source | destination} 255.255.255.255

Switch(config-slb-sfarm)# no nat server

Switch(config-slb-sfarm)# probe probe-name

定义一个虚拟服务器来处理发往服务器群的流量：

Switch(config-module-csm)# vserver virtual-server-name

Switch(config-slb-vserver)# serverfarm serverfarm-name

Switch(config-slb-vserver)# virtual ip-address [network-mask] any

Switch(config-slb-vserver)# vlan vlan-number

Switch(config-slb-vserver)# inservice

Switch(config-slb-vserver)# replicate csrp {sticky | connection}

定义一个通用服务器群处理离开防火墙群的流量：

Switch(config-module-csm)# serverfarm serverfarm-name

Switch(config-slb-sfarm)# predictor forward

Switch(config-slb-sfarm)# no nat server

定义一个通用的虚拟服务器处理离开防火墙群的流量：

Switch(config-module-csm)# vserver virtual-server-name

Switch(config-slb-vserver)# serverfarm serverfarm-name

Switch(config-slb-vserver)# virtual 0.0.0.0 0.0.0.0 any

Switch(config-slb-vserver)# vlan vlan-number

Switch(config-slb-vserver)# inservice

7.3 配置 CSS 实现

配置 CSS 的物理接口：

(config) interface interface_name

(config-if) bridge vlan vlan-id (或者(config-if) trunk)

指定 IP 地址：

(config) circuit circuit_name

(config-circuit) ip address ip_address subnet_mask

(config-circuit-ip) enable

配置缺省路由：

(config) ip route 0.0.0.0 0.0.0.0 next-hop-address

定义防火墙群的防火墙：

(config) ip firewall index local_firewall_address remote_firewall_address remote_css_address

定义静态路由：

(config) ip route ip_address subnet_mask firewall index distance

调整 Keeplive 时间：

(config) ip firewall timeout seconds

验证命令 show ip firewall 防火墙状态，show ip routes firewall 到防火墙的静态路由，show flows 显示到防火墙的负载均衡连接。

第八章　日志管理

8.1 时钟管理

定义时区：

Firewall(config)# clock timezone zone-name hours [minutes]

定义夏令时：

Firewall(config)# clock summer-time zone recurring [week weekday month hh:mm week weekday month hh:mm] [offset]

Firewall(config)# clock summer-time zone date {day month | month day} year hh:mm {day month | month day} year hh:mm [offset]

设置防火墙时钟：

Firewall(config)# clock set hh:mm:ss {day month | month day} year

时钟验证：

Firewall# show clock [detail]

指定 NTP 服务器：

Firewall(config)# ntp server ip-address [key number] [source if-name] [prefer]

配置 NTP 认证：

Firewall(config)# ntp authentication-key key-number md5 value

Firewall(config)# ntp trusted-key key-number

Firewall(config)# ntp authenticate

NTP 验证：show ntp, show ntp status, show ntp associations

8.2 日志配置

启用消息日志：

Firewall(config)# logging on (7.x 用 logging enable)

使用事件列表定义日志策略 (7.0 特性)：

Firewall(config)# logging list event_list level level [class event_class]

Firewall(config)# logging list event_list message start[-end]

根据不同日志级别定义目的位置 (7.0 特性)：

Firewall(config)# logging class event_class destination level [destination level] [destination level] ...

发送日志到 console：

Firewall(config)# logging console level

发送日志到 telnet，ssh 会话：

Firewall(config)# logging monitor level

发送日志到 buffer：

Firewall(config)# logging buffered level

发送日志到 ftp (7.0 特性)：

Firewall(config)# logging ftp-bufferwrap

Firewall(config)# logging ftp-server ftp_server path username password

发送日志到 flash (7.0 特性)：

Firewall(config)# logging flash-bufferwrap

Firewall(config)# logging flash-minimum-free kbytes_free

Firewall(config)# logging flash-maximum-allocation kbytes_max

发送日志到 SNMP 服务器：

Firewall(config)# snmp-server host [if_name] ip_addr trap

Firewall(config)# snmp-server host if_name ip_addr TRap [community string] [version version] [udp-port port] (7.x)

Firewall(config)# snmp-server enable traps {all | syslog}

Firewall(config)# logging history level

发送日志到 Syslog 服务器：

Firewall(config)# logging trap level

Firewall(config)# logging device-id {hostname | ipaddress if_name | string text}

Firewall(config)# logging host if_name ip_address [protocol/port] [format emblem]

Firewall(config)# logging timestamp

Firewall(config)# logging queue queue_size (show logging queue 验证)

Firewall(config)# logging facility facility

Firewall(config)# logging standby

发送日志到邮件 (7.x 特性)：

Firewall(config)# logging mail {level | event-list}

Firewall(config)# smtp-server server_primary [server_secondary]

Firewall(config)# logging from-address from_email_address

Firewall(config)# logging recipient-address to_email_address [level level]

发送日志到 ASDM：

Firewall(config)# logging asdm-buffer-size num_of_msgs

Firewall(config)# logging asdm {level | event-list}

验证 show logging

8.3 日志消息输出的微调

消息的修剪：

Firewall(config)# no logging message message-number (show logging message 验证)

改变消息严重等级：

Firewall(config)# logging message message-number [level level]

配置日志对 ACL 支持：

Firewall(config)# access-list acl_name {permit | deny} ... log [level] [interval seconds]

Firewall(config)# access-list deny-flow-max n

Firewall(config)# access-list alert-interval seconds

8.4 日志分析

对日志分析的软件

CS-MARS (http://www.cisco.com)

Network Intelligence Engine (http://www.network-intelligence.com)

Network Security Analyzer 和 FirewallAnalyzer Enterprise (http://www.eiqnetworks.com)

Sawmill Log Analyzer (http://www.sawmill.net)

CiscoWorks (http://www.cisco.com)

第九章　防火墙工作状态验证

9.1 防火墙健康检查

CPU 负荷：

Firewall# show cpu usage (show cpu usage context all 正常应该在 80%以下)

Show processes 显示防火墙当前活动进程，一般关注 Process 和 Runtime。

内存利用：

Firewall# show memory

Xlate 表大小：

Firewall# show xlate count

Conn 表大小：

Firewall# show conn count

防火墙流量 使用 PDM，Syslog, show traffic 来计算或者 Perfmon 计数器：

Firewall# show perfmon

Firewall(config)# perfmon interval seconds ,perfmon {verbose | quiet}

Inspection 引擎和 Service Policy：

Firewall# show service-policy

Failover：Firewall# show failover

端口状态：

Firewall# show interface

包队列状态：

Firewall# show priority-queue statistics [if_name]

9.2 流经防火墙数据的监控

特性介绍　对于流经防火墙数据的监控有两种方式 capture session 和 debug

packet，两者区别在于前者可以后处理，多个进程，CPU 和内存利用率低，后者是实时显示，同时只能一个进程，且对资源利用率高，后者在 7.x 后已经不被支持。

配置 Capture

配置兴趣流量的 ACL：

Firewall(config)# access-list acl_id [line line-num] [extended] permit protocol {source_addr source_mask [operator sport] [destination_addr destination_mask [operator dport]]

配置 Capture：

Firewall# capture capture_name [access-list acl_name] [ethernet-type type] [interface if-name] [buffer bytes] [circular-buffer] [packet-length bytes] (7.x 支持 type {raw-data | isakmp | asp-drop drop-reason}参数)

show capture 显示当前的 Capture 会话：

Firewall# show capture capture_name [access-list acl_name] [detail] [dump] 显示所抓包的信息。

Firewall# copy capture:capture-name tftp://server/path [pcap] 拷贝信息至 TFTP，如果启用 http 后可以用 https://firewall_address/capture/capture_name[/pcap]通过 Web 来显示或者下载。

clear capture capture_name 清空 capture 缓存但是保持会话

no capture capture_name interface if_name 停止 capture，从特定接口去除保持会话和缓存，no capture capture_name 彻底删除会话和缓存。

配置 Debug 模式：

Firewall# debug packet if_name [src source_ip [netmask mask]] [dst dest_ip [netmask mask]] [[proto icmp] | [proto {tcp | udp} [sport src_port] [dport dest_port]] [rx | tx | both]

9.3 验证防火墙的连接性

Ping 测试：

Firewall# ping [if_name] host [data pattern] [repeat count] [size bytes] [timeout seconds] [validate]

ARP 缓存检查：

show arp

路由表检查：

show route

Traceroute 测试：

traceroute 命令前提配置

Firewall(config)# access-list acl_name permit icmp any any eq echo

Firewall(config)# access-list acl_name permit icmp any any eq echo-reply

Firewall(config)# access-list acl_name permit icmp any any eq unreachable

Firewall(config)# access-list acl_name permit icmp any any eq time-exceeded

Firewall(config)# access-list acl_name permit udp any range 32768 65535 any range 33434 33523

Firewall(config)# access-list acl_name permit udp any dns_address eq domain (可选)

ACL 检查：

show access-group, show access-list

NAT 验证：

Firewall# show xlate [detail] [global | local ip1[-ip2] [netmask mask]] lport | gport port[-port]] [interface if1[,if2][,ifn]] [state static [,dump] [,portmap] [,norandomseq] [,identity]] [debug] [count]

Firewall# show xlate [{global | local} ip1[-ip2] [netmask mask]] [{lport | gport} port[-port]] [interface if1[,if2][,ifn]] [state {static | portmap | identity | norandomseq}] [debug] [detail]

Firewall# show conn [state state_type] [{foreign | local} ip1[-ip2] netmask mask] [long] [{lport | fport} port1[-port2]] [protocol {tcp | udp}]

监控特定主机：

Firewall# show local-host [ip_address] [all] [detail]

Firewall# clear xlate global global_ip [netmask mask] [gport global_port]

Firewall# clear xlate local local_ip [netmask mask] [lport local_port]

Firewall# clear xlate interface if_name_1[,if_name_2]

Firewall# clear xlate

超时参数：

Firewall(config)# timeout xlate hh[:mm[:ss]]

Firewall(config)# timeout conn hh[:mm[:ss]]

Firewall(config)# half-closed hh[:mm[:ss]]

Firewall(config)# udp hh[:mm[:ss]]

Shun 检查：

show shun， show shun statistics

用户认证检查：

show uauth show url-server stats

配置更新检查 启用 AAA 记录用户命令记录。