

PIX and ASA

Summary and Addendum

Cisco Firewalls History



- Up to 1997, Cisco only offered a Windows-NT based FW, the **Centri-Firewall**
 - ◆ Acquired from Global Internet Software Group
- **Private Internet Exchange (PIX)**
 - ◆ Originally designed by Brantley Coile and John Mayes of Network Translation, Inc. (Acquired in 1995)
 - ◆ OS: Finesse (Fast InterNEt Server Executive), now known as PIX OS
- **Adaptive Security Appliance (ASA)**
 - ◆ Combines functionality from PIX, VPN 3000 series and IDS
 - ◆ Runs "Security Appliance Code" 7.0 and later

PIX/ASA Comparison



- **Gigabit Ethernet**
 - ◆ Starts with PIX 525 and ASA 5520
- **Software 7.0**
 - ◆ Not available with PIX 501, 506E
- **PIX 501**
 - ◆ 10 simultaneous VPN peers
- **PIX 506E**
 - ◆ 25 simultaneous VPN peers, 2 VLANs
- **PIX 515E**
 - ◆ Up to 5 contexts, 25 VLANs, failover
- **PIX 525**
 - ◆ 100 VLANs, 50 contexts, VAC+ included!, 330 Mbit/s cleartext
- **PIX 535**
 - ◆ 150 VLANs, 50 contexts, VAC+ included!, 1.65 Gbit/s cleartext
 - ◆ Hot swappable power supplies, multiple slots (max 14 interfaces)

PIX/ASA Comparison (cont.)



- **ASA 5510**
 - ◆ 10 VLANs (PIX 515: 25 VLANs)
 - ◆ 300 Mbit/s cleartext (similar PIX 525)
 - ◆ AIP-SSM-10 optional
- **ASA 5520**
 - ◆ 25 VLANs
 - ◆ 450 Mbit/s cleartext throughput
 - ◆ AIP-SSM-10 optional
- **ASA 5540**
 - ◆ 100 VLANs
 - ◆ 400 Mbit/s cleartext throughput
 - ◆ 50 contexts
 - ◆ AIP-SSM-20 optional
- **FWSM**
 - ◆ 1000 VLANs
 - ◆ 5.5 GBit/s cleartext throughput
 - ◆ 100 contexts
 - ◆ **No IPS, no VPN**

PIX/ASA Comparison (cont.)



- **ASA flash cards**
 - ◆ disc0 = flash
 - ◆ disc1 is an optional external flash card
- **Interface identifier**
 - ◆ PIX: ethernet0, ethernet1, ...
 - ◆ ASA: GigabitEthernet0/0, ...
 - ◆ **ASA: Management0/0**
 - Dedicated mgmt interface but can also be used as normal data interface
- **Only ASA supports**
 - ◆ **AIP-SSM for enhanced IPS**
 - ◆ **Web-VPN**

VPN Acceleration Card (VAC)



- **Features:**
 - ◆ Fast IPsec tunnel establishment (DH in HW)
 - ◆ 3750 USD list price
 - ◆ Only one VAC per device
- **VAC uses 32 bit / 33 MHz bus**
 - ◆ Only 3DES, no AES
 - ◆ Approximately 3x faster than without VAC
- **VAC+ uses 64 bit / 66 MHz bus**
- **Performance:**
 - ◆ PIX 535 with 3DES
 - VAC: 61-100 Mbit/s
 - VAC+: 265-425 Mbit/s
 - ◆ PIX 535 with AES and VAC+
 - AES-128: 315-495 Mbit/s
 - AES-256: 290-425 Mbit/s

TCP Revisited

This short section is needed to understand:

- **FW's randomization of TCP's initial sequence numbers (ISNs)**
- **SYN-Cookies / Intercept**
- **FW's Connection-Table**



- Three-way handshake
 - ◆ Only needed for ISN synchronization !!!
- TCP's **Window** parameter
 - ◆ = Number of bytes that can be sent without receiving an ACK
 - ◆ Typically between 512 bytes to 32 Kbytes
- Typical Attacks:
 - ◆ **DoS by continuously sending RST**
 - Requires knowledge of socket information (difficult to guess client side – except DNS which uses same ports)
 - Requires knowledge of valid ACK-numbers (must be within window, otherwise not accepted)
 - ◆ **DoS via SYN flood**
 - FW can limit number of embryonic sessions
 - ◆ **Intrusion via session hijacking**
 - Allows to play MITM similar as ARP spoofing in LAN
 - But again valid ACK numbers and socket information needed

TCP Revisited (cont.)



- **TCP security is mainly based on pseudorandom ISNs !!!**
 - ◆ Only if SN within window, packets are accepted
 - ◆ Power of ISN-randomization (e. g. by FW or modern operating systems) is limited:
 - SNs must always be larger than those of previous session!
(Otherwise packets could interfere)
- **FIN Sequence:**
 - ◆ Either "Simultaneous Close" (FINACK-FINACK)
 - ◆ Or "Normal Shutdown Sequence" (RFC 793: FIN, ACK, FIN, ACK)
- **Reasons for a RST:**
 - ◆ Upon receiving packets but there is no session
 - ◆ Upon receiving an ACK but nothing has been sent
 - ◆ Security level problems and FW intervention

TCP Hijacking



- First bring server and/or client in a **desynchronized state**
 - ◆ Incoming packet is dropped if SQNR is either
 - Smaller than expected (local ACK number)
 - Greater than ACK+WIN
 - ◆ Can result in ACK-storm: both parties tell other which SQNR is expected
 - ◆ Desynchronization done via
 - **RST+SYN** (during 3-way handshake)
 - Or by inserting **large payloads** (need to guess SQNRs if sent blindly)
- Then send any packets to server and/or client
 - ◆ E. g. Telnet session: "echo 'hacked you' > hello.txt"

TCP Hijacking (cont.)



- Interesting for attacker:
 - ◆ If server requires IP-based client authentication
 - ◆ Or another authentication has been performed at the very beginning of the session (such as OTPs or similar)
- MITM is possible:
 - ◆ If attacker uses IP's *strict source route* option
 - ◆ Or if attacker resides on LAN and uses ARP spoofing or ICMP redirect
- **Guessing ISN is often simple!**
 - ◆ RFC requires SN to be increased every 4 usec
 - ◆ But practically it is done in long intervals (e. g. every second by 250,000)
 - ◆ **Therefore, additional randomization by firewall needed!**
- Only encryption (IPsec) really helps!

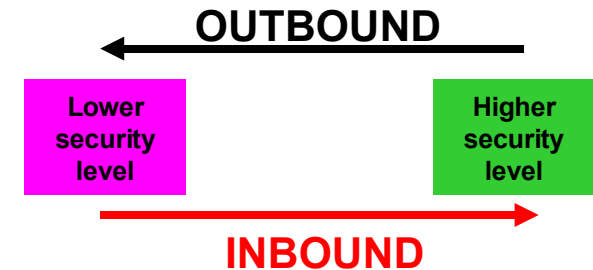
Basic Protection Mechanisms

Note: In the following slides the abbreviation ASA means the Adaptive Security Algorithm and NOT the appliance (which also relies on this algorithm).

Adaptive Security Algorithm (ASA)

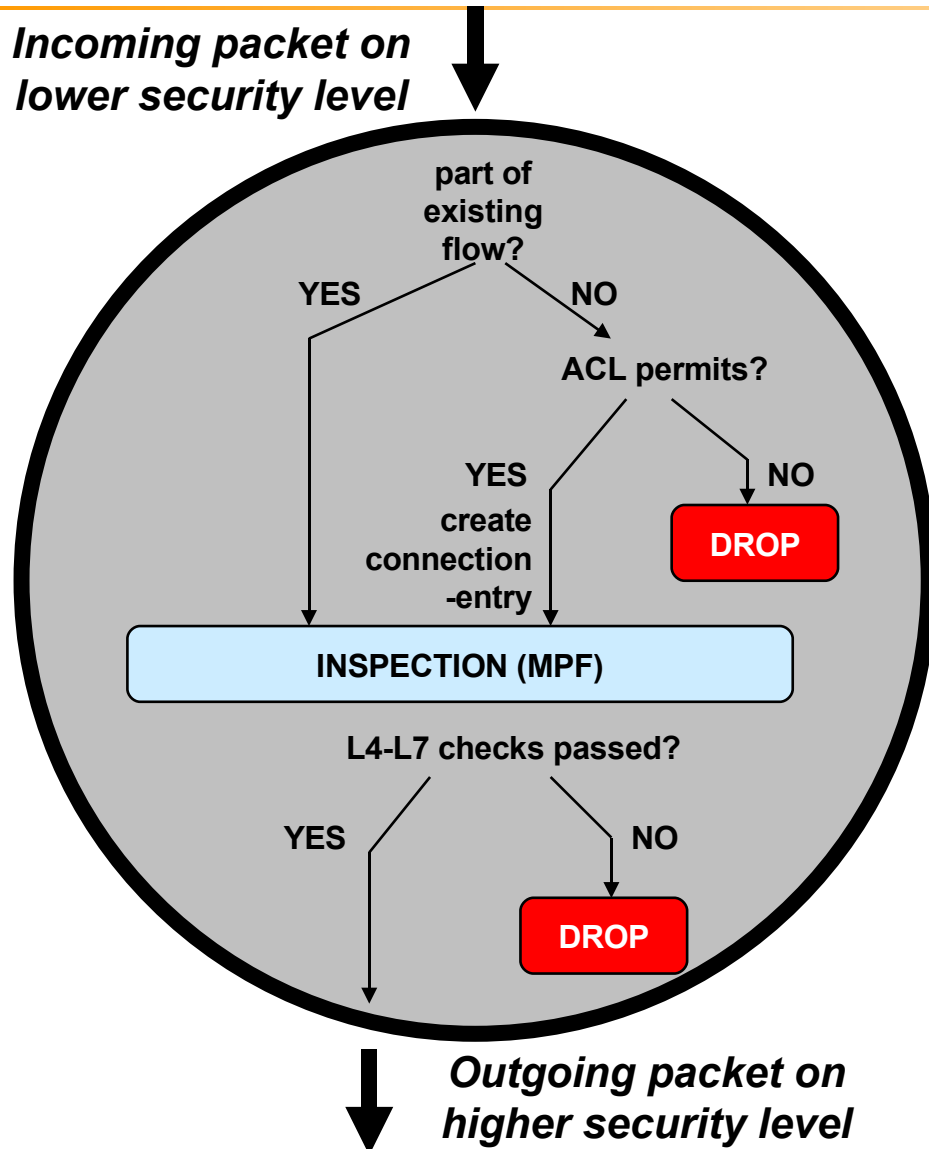


- The **ASA** (don't confuse with the appliance) **is always on**
 - ◆ Cannot be truly disabled
 - ◆ At least L4 checking is done
- **Default policy based on security levels**
 - ◆ Values 0..100, assigned per interface
 - ◆ Interfaces may have same security level
 - ◆ **Only traffic from higher to lower level is allowed (default policy)**
 - ◆ Exceptions can be configured via ACLs
- **Stateful**
 - ◆ For TCP: ***SQNRS randomized***, Flags checked (default idle timeout: 1 hour)
 - ◆ For UDP: Only timeouts applied (default idle timeout: 2 minutes)
 - ◆ ***For nothing else!***
 - E. g. echo-replies (for outbound echo-requests) are ***filtered*** by default (ACL needed!)





ASA Flowchart



- **Inspect** (old: fixup)
 - ◆ Allow application specific tunings
 - ◆ Most important inspects already enabled
 - ◆ *Also important for NAT*
- **The FW is invisible!**
 - ◆ ToS and TTL is not changed
- **Never route a packet out of the receiving interface!**
 - ◆ Packets must *always* be passed to another interface
 - ◆ Security feature – but limits hub & spoke VPN topologies
 - Solution: see VPN section (same-security-traffic permit intra-interface)



- **Used to define exceptions of ASA's default policy**
 - ◆ E.g. **permit inbound traffic** (from lower-to-higher security level)
 - ◆ E. g. **deny outbound traffic** (from higher-to-lower security level)
- **ACL only needs to describe initial packet**
 - ◆ **No need to think on return traffic** (=> ASA does this)
 - ◆ Therefore ACLs are only checked **once** per connection (then the connection table is used)
 - ◆ Extended and standard ACLs possible
 - Netmask instead of wildcard mask
 - Standard ACLs check **destination** address !!!
- **Delete an access-list from running-config:**
 - ◆ `clear configure access-list ACLIN`



- **Configure anti-spoofing rules**
 - ◆ deny "impossible" source addresses for incoming packets on the outside interface
 - ◆ RFC 1918 and others (e. g. loopback, etc)
 - ◆ These rules are usually NOT generated by security policy tools (such as CSPM)
 - ◆ Preferably “outsource” to adjacent router
- **Use object-grouping**
 - ◆ For simpler configuration and to mitigate misconfigurations
 - ◆ No performance impact !!!
 - ◆ Object-groups are expanded to normal ACLs internally



ACL Logging:

- Use `log` parameter for important ACEs
- Specify a **log interval** to reduce load
 - ◆ After first match of ACE a syslog message is sent (“... hit-cnt 1 first hit...”)
 - ◆ Additional packets are counted but next syslog message is only sent after specified interval
- Send an alert message upon **too many concurrent deny-flows**
 - ◆ Cause: Either an attack or FW is too restrictive

```
(config)# access-list OUTSIDE permit icmp host defgw  
192.168.1.0 255.255.255.0 log interval 300
```

```
! Deny-flox-max = {1..4096}, alert-interval = {1..3600} sec  
! Here the defaults are used:  
(config)# access-list deny-flow-max 4096  
(config)# access-list alert-interval 300
```

(FYI: Turbo ACLs *now default!*)



```
(config)# access-list [NAME] compiled
```

- A tree of 12-18 levels is created => upper bound on lookup time
 - ◆ Independent of number of ACEs
 - ◆ Short ACLs (< 18 ACEs) are never compiled
- Requires at least 2.1 MByte
 - ◆ Plus approx 1 MByte for every 2000 ACEs (!!!)
- Turbo ACLs are used by default on modern appliances
 - ◆ Such as the FWSM
- Note: Reconfiguring Turbo-ACLs causes a CPU spike
- Introduced in PIX firewall version 6.2
- Version 7.0 automatically optimizes access list processing
 - ◆ No need to configure

ICMP permit | deny <IP> <icmp-type> <if>



- For traffic ***terminating*** at a FW interface
- Default:
 - ◆ Allow traffic terminating on FW interfaces
 - ◆ Even echo request, even on outside interface
 - ◆ Except when DA=broadcast



- **NAT configuration not required (since version 7.0)**
 - ◆ However 'address-hiding' is recommended
 - ◆ Use the `nat-control` command to make NAT configurations mandatory again (as before version 7.0)
- **NAT configurations:**
 - ◆ Dynamic NAT: `nat (inside)` and `global (outside)` commands
 - ◆ Static NAT:
`static (inside,outside) <global-ip> <local-ip>`
 - ◆ NAT Exemption: `nat (inside) 0 <ip> <mask>`
- **Note:**
 - ◆ Packet precedence: `nat 0` => static NAT => dynamic NAT
 - ◆ Static NAT can also be used for port redirection
 - ◆ PAT is done automatically when only a single global IP address has been specified with the `global` command

The `filter` Command



- **Protect inside hosts:**
 - ◆ From downloading malicious code (Java, ActiveX)
 - ◆ From accessing unwanted content (url, https, ftp) specified by an URL server
- **`(config)# filter url http 0 0 0 0 allow`**
 - ◆ `0 0 0 0` = any IP to any IP
 - ◆ Allow = fail open (when URL server fails)
- **For URL(HTTP), HTTPS, and FTP filtering, an URL server must be specified:**
 - ◆ `(config)# url-server (dmz) vendor websense
host 172.16.1.254 timeout 20 protocol TCP
version 4`

Old: Fixup => New: Inspect



- **Stateful application inspection**
 - ◆ To ensure secure use of applications and services
 - ◆ Important to allow return traffic for applications with dynamic port negotiation
- **Fixups**
 - ◆ Up to version 6.3
- **Inspect**
 - ◆ Since version 7.0, the fixup command has been deprecated and is automatically replaced with the **inspect** command
 - ◆ Uses the Modular Policy Framework (MPF) infrastructure (see next section)

Fragment Handling



- < 4.2
 - ◆ Check only offset values
- 4.2-5.0
 - ◆ "Fragguard" feature
 - Only *total number* of fragments (in a chain) limited
- **Since 5.1**
 - ◆ Virtual reassembly buffers
 - ◆ PIX reassembles and checks packet before forwarding
 - ◆ Enabled by default

```
! Size of reassembly buffer in number of packets (max 10^6):  
(config)# fragment size 200  
  
! Specify maximum allowed number of fragments per packet:  
(config)# fragment chain 24  
  
! Specify reassembly timeout (max seconds to wait for whole packet):  
(config)# fragment timeout 5  
  
!!! Prevent incoming fragmented packets at all: !!!  
(config)# fragment chain 1 outside
```

(specified values are the defaults)

Sysopt – Define Exceptions



- **Allows certain trusted packets to bypass ASA and ACLs**
 - ◆ For example useful for IPsec, L2TP, PPTP
- **Tweaks default handling for certain protocols**

Most important:

```
! Let VPN traffic bypass ACLs:  
(config)# sysopt connection permit-vpn  
! Set maximum MSS for TCP (also a minimum MSS can be set):  
(config)# sysopt connection tcpmss 1380  
! Wait 15 seconds after a simultaneous FIN to prevent too  
! many CLOSING states in certain operating systems:  
(config)# sysopt connection timewait
```

Also interesting:

```
! Disable DNS record alteration:  
(config)# sysopt nodnsalias inbound  
! disable proxy ARP for NAT global addresses on an interface:  
(config)# sysopt noproxyarp outside  
! Some Radius-servers do not use key in hash of accounting-ACKs  
! PIX would repeat accounting-RQ endlessly...  
(config)# sysopt radius ignore-secret
```



sysopt connection tcpmss

```
(config)# sysopt connection tcpmss 1200
(config)# sysopt connection tcpmss minimum 500
```

- The PIX/ASA alters the MSS-option announced in the 3-way handshake
 - ◆ MSS too large => rewrite with maximum MSS
 - ◆ MSS too small => rewrite with minimum MSS
- Note: The *payload size* is specified !!!
 - ◆ 1380 bytes is the default and is the worst case for a 1500 byte MTU:
 - 20 bytes outer IP
 - 24 bytes AH + 24 bytes ESP + 12 bytes ESP_AUTH
 - 20 bytes inner IP + 20 bytes TCP
 - => up to 1380 bytes left for payload



sysopt connection timewait



- **PIX/ASA would close TCP connection slots upon receipt of FIN**
 - ◆ Retransmission after a FIN would cause denied packets
 - ◆ Especially important with a "simultaneous close" and certain operating systems
 - Their sockets might linger in a closing state (waiting for an ACK)
- **Using sysopt connection timewait**
 - ◆ Keeps state for additional 15 seconds
- **It can be basically recommended !**
 - ◆ Pro: Cleans logs, helps certain OSes
 - ◆ Cons: Degrades performance when number of connections is very high

The *established* command



- If an outbound connection to a particular host exists, then also *certain* inbound traffic is allowed
- *Does not work with PAT*

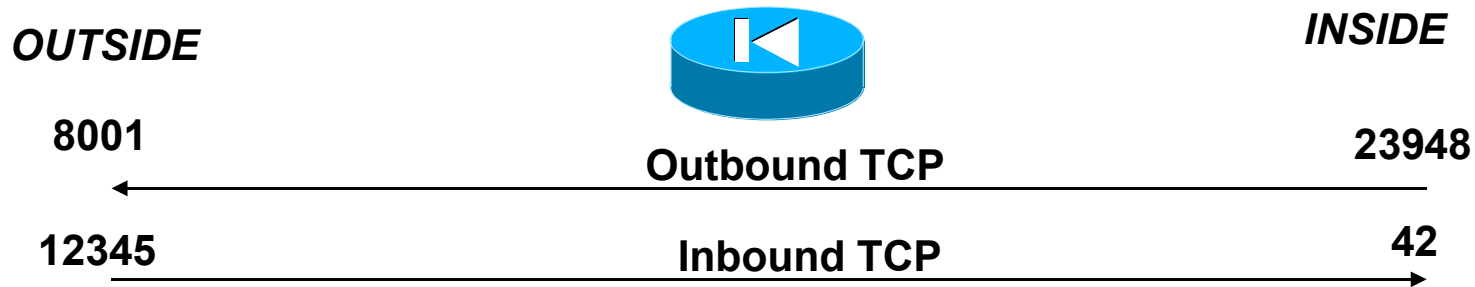
```
established protocol [srcport] dstport permitfrom protocol  
port-port permitto protocol port-port
```

Or in short:

```
established P1 A B permitfrom P2 E permitto P2 F
```

Which reads, "if there *exists* a connection between two hosts using **protocol P1** from **srcport A** to **dstport B**, *permit return* connections through the firewall via **protocol P2** (which can be different from P1), if the **source port(s)** correspond to **E** and the **destination port(s)** correspond to **F**."

Examples



Security hole:

```
established tcp 0 8001
```

After internal host connects to port 8001 on the external host, the external host can enter the inside network using any protocol and any port number. (0=any port)

Better:

```
established tcp 8001 permitto tcp 42
```

Much better:

```
established tcp 8001 permitfrom tcp 12000-13000 permitto tcp 42
```

Best: be as restrictive as possible!

```
established tcp 23948 8001 permitfrom tcp 12345 permitto tcp 42
```



- Provides additional anti-spoofing protection
 - ◆ Disabled by default
 - ◆ Can be safely enabled on every interface (no asymmetric routing since PIX is stateful)
- Ingress and egress verification
 - ◆ In both cases the "arriving interface" must be the best according to the local routing table
- Note: Always have a [static default route](#) in the routing table, if Internet connections are supported
 - ◆ Otherwise, if the source network is not matched by a routing entry, the packet is dropped !!!

```
# ip verify reverse path interface if_name
```



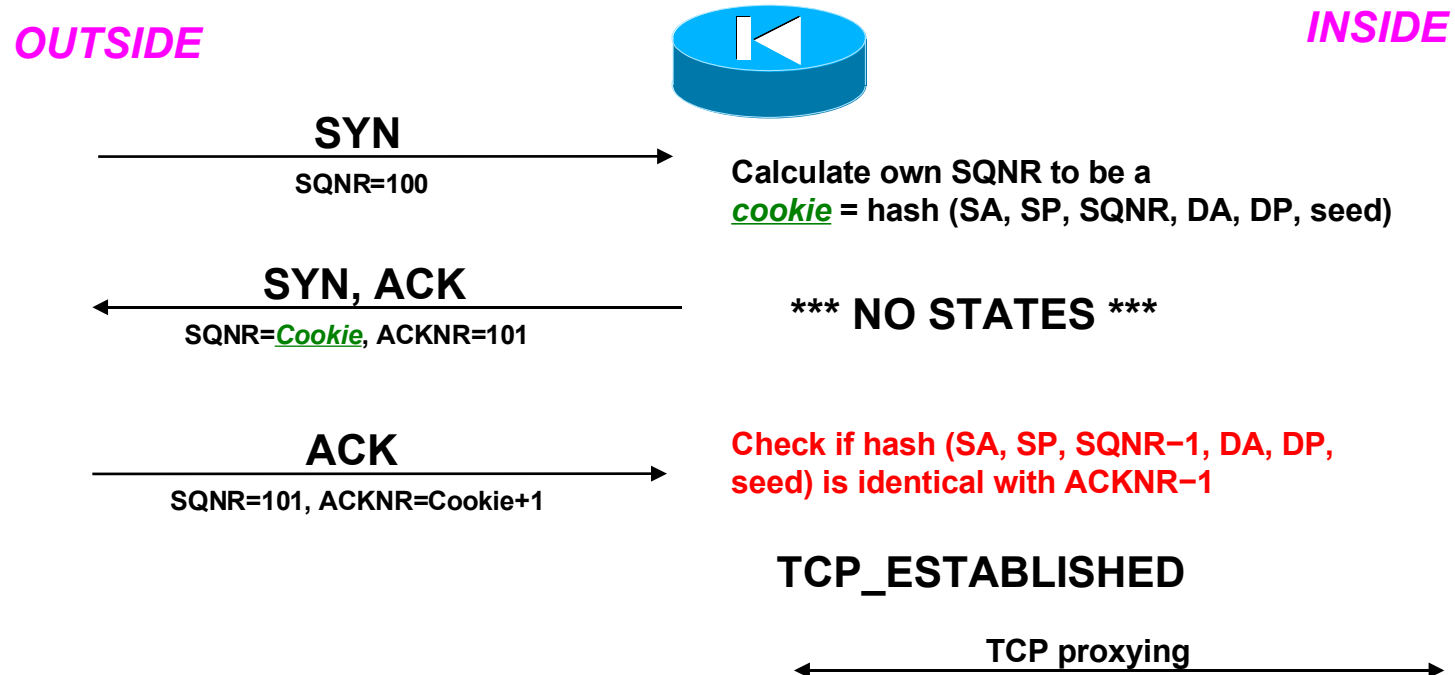
- Especially recommended if there is something “useful” to check
 - ◆ E. g. if FW uses multiple interfaces and has lots of routing information (not only a default route)
 - ◆ Not much sense if FW has only one outside IF connected to the Internet
- Note:
 - ◆ For UDP and TCP only the initial packet is checked
 - ◆ But ICMP packets have no session parameters, so **every** packet is checked
 - **Could be abused by attacker for FW-DoS**

TCP-Syn Attacks: Intercept + Cookies

- Old: **TCP Intercept**
 - ◆ When number of embryonic TCP connections per host exceeds a certain SYN threshold,
 - ◆ Then PIX *intercepts* inbound TCP three-way handshakes and awaits ACK
 - ◆ Until number of embryonic sessions falls below a minimum threshold
 - ◆ *Very CPU intensive !!!*
- New: **SYN Cookies**
 - ◆ Requires version 6.2.x or later
 - ◆ Also controls number of embryonic TCP connections per host but works stateless
 - ◆ Fully replaced intercept feature !!!



- **More scalable than TCP Intercept**
 - ◆ No states needed on FW
 - ◆ **Cookie = hash (sockets, SQNR, secret)**



SYN Flood Protection Configuration



- **Enable SYN-Cookies by specifying an `em_limit`**
 - ◆ Specifies maximum embryonic connections *per host*
 - ◆ An `em_limit` value of 0 means: unlimited number of embryonic connections allowed
- Can be configured as parameter in the `static` or `nat` command, or via a policy map
- Also consider specifying a **connection limit** per host using the **`conn_limit`** parameter

```
static (in_if, out_if) mapped_addr real_addr [netmask mask]
[conn_limit [em_limit]]
```

```
! Recommended: start intercepting/SYN-cookies immediately:
(config)# static (inside,outside) 1.2.3.4 10.0.0.1 0 1
```

NOTE:

- TCP-Intercept scales up to several thousands of SYNs per second (~ 1-2 Mbit/s)
- Of course also SYN-cookies require some CPU resources – but significantly less!



- Can later be analyzed with Ethereal
- Default: only a packet-length of 68 bytes are copied

```
(config)# capture MY_CAP_FILE access-list ACL_CAP
interface outside
# show capture MY_CAP_FILE [detail]
# copy capture:MY_CAP_FILE tftp:
# copy capture:MY_CAP_FILE https://10.0.0.1/sub/pcap
```

- Also available on FWSM

Address Shunning



- **Quickly block traffic from a specified source address**
 - ◆ **No ACL or interface specification needed**
 - ◆ **Typically used in emergency cases only**

```
shun src_ip [dst_ip src_port dest_port [protocol]] [vlan vlan_id]
```

Examples:

```
! Block all traffic from that offending host 141.45.54.9
# shun 141.45.54.9

! Block a specific connection:
# shun 141.45.54.9 10.0.8.11 33445 8080 tcp
```

Modular Policy Framework

Modular Policy Framework (MFP)



```
(config)# class-map MYMAP
(config-map)# match access-list XYZ
(config)# class-map SOME
(config-map)# match dscp 5
(config)# class-map OTHER
(config-map)# match default-inspection-traffic

(config)# policy-map MYPOL
(config-pmap)# class MYMAP
(config-pmap-c)# set connection random-sequence-number disable
(config-pmap-c)# inspect icmp
(config-pmap-c)# ips inline fail-close
(config-pmap)# class SOME
(config-pmap-c)# priority
(config-pmap)# class OTHER
(config-pmap-c)# police input 128000 2000 conform-action transmit
                    exceed-action drop
(config)# service-policy MYPOL global | interface outside
```

Cut-Through Proxy

Proxy Authentication (“Cut Through”)

- **Process details with HTTP:**
 1. The PIX *intercepts* HTTP request from the user
 2. PIX sends authentication request to AAA server
 3. Authentication response contains the *ACL-name* (as configured on the AAA server)
 4. PIX checks whether user’s ACL is already cached
 5. *If not, PIX requests ACL from AAA server*
 6. Finally the HTTP request is forwarded
- ◆ **Also FTP, HTTPS and Telnet supported**

Configuration



- **On ACS: Define PIX as AAA client and configure users**
- **On PIX: Define ACS, ACL for interesting traffic and enable cut-through authentication via `aaa authentication match`**

```
! Define AAA server
(config)# aaa-server MYACS protocol tacacs+
(config)# aaa-server MYACS host 10.1.1.200
(config-aaa-server-host)# key SeCrEtKeY

! Define interesting traffic which requires authentication:
(config)# access-list PROX permit tcp any host 192.168.1.254 eq ftp
(config)# access-list PROX permit tcp any host 192.168.1.254 eq http
!
! Enable cut-through proxy:
(config)# aaa authentication match PROX inside MYACS
!
! Specify timeouts (defaults shown)
(config)# timeout uauth 0:05:00 absolute
(config)# timeout uauth 0:00:00 inactivity

! Optionally enforce HTTPS-based user authentication (recommended)
(config)# aaa authentication secure-https-client
```

Virtual Authentication



- **Virtual Telnet**
 - ◆ Needed when services without interactive user session (i. e. other than Telnet, FTP, or HTTP) requires authentication (e. g. VoIP)
 - ◆ Then clients must first authenticate at the PIX via Telnet
 - ◆ Disadvantages
 - Not user friendly - prefer web-based authentication instead
 - An unused address must be used (from global pool if outside auth)
- **Virtual HTTP**
 - ◆ When proxy authentication is used, and the HTTP server also requires authentication
 - Normally, same username and password you used to authenticate with the security appliance is sent to the HTTP server; you are not prompted separately for the HTTP server username and password
 - ◆ Solution is Virtual HTTP:
 - PIX sends a HTTP redirect to the client
 - Client's browser will establish a new HTTP connection and does not use cached credentials
 - ◆ Since version 7.2 (or so) it is now a default on the PIX/ASA [it seems!]

```
(config)# access-list VAUTH permit tcp any host 10.1.1.111
(config)# aaa authentication match VAUTH inside MYACS
(config)# virtual telnet 10.1.1.111
```



- **Problem:**
 - ◆ If only authentication configured then user may do anything which is not explicitly denied by interface ACL or default ASA policy
 - ◆ Therefore use group- or user-ACLs for specific authorizations (also to allow additional services)
- **Per-user (or per-group) ACLs**
 - ◆ Only supported with RADIUS
 - ◆ Authentication response contains ACL
 - ◆ ACL is applied on interface where user "comes in"
- **If either downloaded ACL or interface ACL has a deny then the deny holds**
 - ◆ Option: `access-group` command allows additional keyword `per-user-overwrite`
 - Then downloaded ACL overwrites interface ACL



- **Shared Profile Components**
 - ◆ Specify ACL name, description, and ACL definitions (=rules)
- **User Setup**
 - ◆ Under “Downloadable ACLs” check “Assign IP ACL“ and select defined ACL-name
- **Group Setup**
 - ◆ If ACL should be forwarded to a group
 - ◆ Same as with User Setup

AAA Config



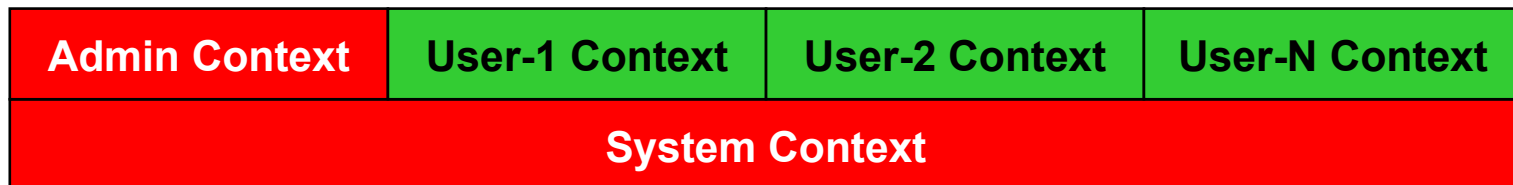
```
(config)# aaa-server MY_ACS protocol tacacs+
(config-aaa-server-group)# accounting-mode single | simultaneous
(config-aaa-server-group)# max-failed-attempts <1-5>
(config-aaa-server-group)# reactivation-mode depletion | timed !30s
!
(config)# aaa-server MY_ACS (inside) host 10.1.1.200
(config-aaa-server-host)# key SeCrEtKeY
(config-aaa-server-host)# timeout 10 !retransmission timeout
```

Security Contexts

Virtualization: Security Contexts



- **Mandatory contexts:**
 - ◆ **System Context:** uses startup-config and defines all other contexts
 - ◆ **Admin Context:** to configure network connectivity for management, AAA, ...
- **Restrictions**
 - ◆ All contexts are either in transparent or routed FW mode
- **Unsupported Features for contexts:**
 - ◆ VPN !!!
 - ◆ **Routing Protocols** (use only static routes)
 - ◆ **IP Multicast**



Details: System and Admin Context



■ System context

- ◆ Contains system configuration with basic settings:
 - Which subinterfaces (=>no shut here) and associated VLANs
 - Which contexts: admin / user1 / user2 / ...
 - Allocated interfaces for each context
 - Config-url (=filename) for each context's startup-config
- ◆ Has no connection to network !!!
 - (Uses admin context for communication (syslog, tftp, ...))

■ Admin context

- ◆ Has connection to network !!!
 - IP addresses defined here (for the admin context allocated interfaces only)
 - Security-levels
 - Nameif
 - Also uses [no] shutdown commands per interface !!!
 - ssh / telnet config for the admin
- ◆ Can forward traffic like any other context !!!
 - E. g. via nat / global or static commands



- **If one context per (sub-) interface**
 - ◆ This is the trivial case: Each context has its own dedicated interfaces assigned to
 - ◆ This is also mandatory for transparent FW mode
- **If multiple contexts per (sub-) interface:**
Look at the destination address (DA)
 - ◆ DA matches **interface address** of context?
 - Then the packet (e. g. PING) should terminate here
 - ◆ DA matches **global address of an xlate entry**
 - Either in a static NAT table (static command)
 - Or in a global pool (global command)
 - Classifier relies on translation !!!
 - If no NAT wanted then use a “standalone” global command



mode multiple

- ◆ What happens:

1. Original running-config is saved as old_running.cfg
2. New startup-config is created, used for system context only
3. Admin.cfg is created

mode single

- ◆ Note:

Before reverting back to single mode, install a valid (i. e. single-mode) startup-config file first, because the current configuration is only working for multi-mode !!!

Allocate Interfaces



■ Visible

- ◆ Allows context users to see physical interface details in the show interface command even if you set a mapped name

■ Invisible

- ◆ Allows context users to only see the mapped name (if configured !!!) in the show interface command
- ◆ This is the default !!!

```
(config-ctx)# allocate-interface physical_interface [map_name] [visible | invisible]
```

This is configured and stored in the system-context

Example Configuration



1) System Configuration

```
hostname MyOneAndOnly
password geheim
enable password supergeheim
!
admin-context admin
!
interface gigabitethernet 0/0
  shutdown
interface gigabitethernet 0/0.3
  vlan 3
  no shutdown
!
interface gigabitethernet 0/1
  no shutdown
interface gigabitethernet 0/1.4
  vlan 4
  no shutdown
  interface gigabitethernet 0/1.5
  vlan 5
  no shutdown
  interface gigabitethernet 0/1.6
  vlan 6
  no shutdown
```

```
context admin
  allocate-interface gigabitethernet 0/0.3
  allocate-interface gigabitethernet 0/1.4
  config-url disk0://admin.cfg
context customerA
  description This is the context for customer A
  allocate-interface gigabitethernet 0/0.3
  allocate-interface gigabitethernet 0/1.5
  config-url disk0://context_a.cfg
context customerB
  description This is the context for customer B
  allocate-interface gigabitethernet 0/0.3
  allocate-interface gigabitethernet 0/1.6
  config-url disk0://context_b.cfg
```

Example Configuration (cont.)



2) Admin Context

changeto context admin

```
hostname Admin
domain isp
!
interface gigabitethernet 0/0.3
  nameif outside
  security-level 0
  ip address 192.168.3.254 255.255.255.0
  no shutdown
interface gigabitethernet 0/1.4
  nameif inside
  security-level 100
  ip address 172.16.14.254 255.255.255.0
  no shutdown
!
route outside 0 0 192.168.3.99 1
nat (inside) 1 172.16.14.0 255.255.255.0
global (outside) 1 192.168.3.10-192.168.3.19
!
password GodOnlyKnows
enable password DevilMayCare
ssh 172.16.14.75 255.255.255.255 inside
```

Example Configuration (cont.)



3) Customer A Context

```
interface gigabitethernet 0/0.3
  nameif outside
  security-level 0
  ip address 209.165.201.3 255.255.255.224
  no shutdown
interface gigabitethernet 0/1.5
  nameif inside
  security-level 100
  ip address 10.1.2.1 255.255.255.0
  no shutdown
route outside 0 0 209.165.201.1 1
...
```

Context Example Config



```
hostname(config)# admin-context administrator
hostname(config)# context administrator
hostname(config-ctx)# allocate-interface gigabitethernet0/0.1
hostname(config-ctx)# allocate-interface gigabitethernet0/1.1
hostname(config-ctx)# config-url flash:/admin.cfg

hostname(config-ctx)# context test
hostname(config-ctx)# allocate-interface gigabitethernet0/0.100 int1
hostname(config-ctx)# allocate-interface gigabitethernet0/0.102 int2
hostname(config-ctx)# allocate-interface gigabitethernet0/0.110-gigabitethernet0/0.115
int3-int8
hostname(config-ctx)# config-url ftp://user1:passw0rd@10.1.1.1/configlets/test.cfg

hostname(config-ctx)# context sample
hostname(config-ctx)# allocate-interface gigabitethernet0/1.200 int1
hostname(config-ctx)# allocate-interface gigabitethernet0/1.212 int2
hostname(config-ctx)# allocate-interface gigabitethernet0/1.230-gigabitethernet0/1.235
int3-int8
hostname(config-ctx)# config-url ftp://user1:passw0rd@10.1.1.1/configlets/sample.cfg
```

Interesting Commands



```
! Save configs of all contexts at the same time  
# write memory all
```

Failover Solutions



- Pre 7.0: **active/standby**
 - ◆ Hot standby only
- Since 7.0: **active-active**
 - ◆ Hot standby + load sharing
 - ◆ Requires **multiple contexts** (*no VPN possible !!!*)
- Both methods support optionally **stateful failover**
 - ◆ Synchronizes conn-table
 - ◆ Xlate table is always synchronized (also on not-stateful)
- Either **LAN-based** failover (eth) or **serial cable**
- **Basic principle of failover:**
 - ◆ Standby unit takes over MAC and IP of failed unit

Failover Basics (cont.)



- **Failover license**
 - ◆ Significantly cheaper than normal software license
 - ◆ Cannot be used for a “normal” (active) PIX, because it reboots every 24 hours if the active FW is not seen
- **Use identical machines!**
 - ◆ Same model number
 - ◆ Pre version 7.0: identical OS version
 - ◆ Same activation keys (selects DES or 3DES)
 - ◆ Same amount of RAM and flash
 - ◆ Proper licensing
- **Backup all activation keys if you must recover an IOS image**

LAN-based failover



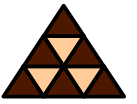
- **Since version 6.2, IP protocol 105**
- **Options**
 - ◆ **Either via crossover cable – Since 7.0 not recommended !!!**
 - Interface on the failover PIX should remain no-shutdown [philosophy: interface state should not depend on the state of the other FW for which failover is configured.]
 - [Also more difficult for the algorithms to find out if self or other machine had a fault (?)]
 - ◆ **Or via a switch in between!**
 - Problem: switch should never go down (FW-SW=SW-FW)
 - ◆ **Encryption**
 - Pre-shared keys (128 bit hex or string)
 - ◆ **Stateful failover**
- **Interfaces should not have a name assigned**
- **Active-Active:**
 - ◆ **FO-link is in system context**

Cable-based failover (only PIX, not ASA)



- **No configuration necessary on failover PIX!**
 - ◆ All configuration (IP address etc) done on primary PIX
- **Uses special serial interfaces and a 2 m cable**
 - ◆ One end of the cable is labelled “Primary”
 - ◆ This serial interface remains up!
 - Therefore more reliable as the LAN-based solution
 - But slower configuration replication
- **No encryption possible**
- **Cable supports 115 kbit/s only**

Stateful Failover (optional)



- **Additional Ethernet connection needed !!!**
 - ◆ With at least 100 Mbit/s !!!
- **Three configuration options:**
 - ◆ 1) Either a **dedicated Ethernet** interface
 - ◆ 2) Or if LAN-based failover is used, the failover link can be **shared** for stateful FO
 - ◆ 3) Or using a regular data interface, such as the inside interface – not recommended!
 - Bad performance
 - Security hole (e. g. replay attacks)



- **Failover (interface) conditions**
 - ◆ If a link is more than **30 sec** down (**hello**s are sent every **15 seconds over all links to each other!**)
 - ◆ **Network activity test fail**: can packets be received during a given period?
 - ◆ **ARP test**: an ARP request is sent to the 10 most recently acquired hosts—at least one should answer
 - ◆ **Broadcast ping test**: any answer within 5 seconds?
- **Other failover conditions**
 - ◆ More than **15 sec** memory exhaustion
 - ◆ When command **failover active** is entered on the **standby** FW
- **After the failover:**
 - ◆ Interface **IP and MAC** addresses are reversed between primary and secondary (except the failover-interfaces)

Failover Details (cont.)



- **If serial cable used**
 - ◆ Cable selects whether PIX is primary or secondary (be careful)
- **Replication of the configuration is done when:**
 - ◆ Standby FW has **booted**
 - ◆ **Every time commands are entered** on the primary FW (hmmm...)
 - ◆ By entering **write standby** on the primary
 - Only into running-config of secondary
- **VPN failover**
 - ◆ Only with Active/Standby



- **Multiple security contexts required!**
- **Divide the security contexts into failover groups**
 - ◆ A failover group is simply a logical group of one or more security contexts
 - ◆ **Maximum of two failover groups**
 - ◆ **The admin context** (and any unassigned security contexts) is always a **member of failover group 1**

Serial Cable Failover



```
! Cable-based Failover ONLY requires configuration of PRIMARY PIX
! Assume serial failover cable connected
(config)# failover
(config)# interface ethernet0
(config-if)# ip address 192.168.2.2 255.255.255.0 standby 192.168.2.7
(config)# interface ethernet1
(config-if)# ip address 10.0.2.1 255.255.255.0 standby 10.0.2.7

! Specify hello interval (3-15 seconds possible, 15 is default)
(config)# failover polltime 15
(config)# failover holdtime 45 ! when declared dead

! Specify link for stateful failover (optional)
(config)# failover link ethernet3

! Specify number of defect interfaces before failover condition
(config)# failover interface-policy <1-250>
! Disable health monitoring on certain uncritical interfaces:
(config)# no monitor-interface ethernet4

# show failover
...
```

LAN-based Failover - Primary



```
! ----- PRIMARY PIX -----

! Note: begin with "no failover" before any configuration otherwise
! Problems could occur. Enable failover only afterwards
(config)# no failover

(config)# interface ethernet0
(config-if)# ip address 192.168.2.2 255.255.255.0 standby 192.168.2.7

(config)# interface ethernet1
(config-if)# ip address 10.0.2.1 255.255.255.0 standby 10.0.2.7

(config)# failover lan interface LANFAIL ethernet2
(config)# failover interface ip LANFAIL 172.17.2.1 255.255.255.0
      standby 172.17.2.7

(config)# failover lan enable
(config)# failover lan unit primary
(config)# failover key 1234567
(config)# failover
```

LAN-based Failover - Secondary



```
! ---- SECONDARY PIX -----
```

```
(config)# failover lan interface LANFAIL ethernet2
```

```
(config)# failover interface ip LANFAIL 172.17.2.1 255.255.255.0  
standby 172.17.2.7
```

```
(config)# failover lan unit secondary
```

```
(config)# failover lan key 1234567
```

```
(config)# failover lan enable
```

```
(config)# failover
```

```
Detected an Active mate
```

```
Beginning configuration replication from mate.
```

```
End configuration replication from mate.
```

Active-Active – Primary



```
(config)# mode multiple
...reboot...
(config)# interface ethernet2
(config-if)# no shut
(config)# failover lan interface MYFAILOVER ethernet2
(config)# failover lan enable
(config)# failover interface ip MYFAILOVER 172.16.1.1 255.255.255.0
    standby 172.16.1.7
(config)# failover link MYFAILOVER ethernet2
(config)# failover lan key 1234567
(config)# failover lan unit primary
(config)# failover group 1
(config-fover-group)# primary
(config)# failover group 2
(config-fover-group)# secondary
(config)# context admin
(config-ctx)# allocate-interface ethernet0
(config-ctx)# allocate-interface ethernet1
(config-ctx)# config-url flash:/admin.cfg
(config-ctx)# join-failover-group 1
```

Active-Active – Primary/Secondary



```
(config)# context ctx1
(config-ctx)# allocate-interface ethernet3
(config-ctx)# allocate-interface ethernet4
(config-ctx)# config-url flash:/ctx1.cfg
(config-ctx)# join-failover-group 2
```

===== SECONDARY =====

!!! First finish configuration on PRIMARY and enter write mem !!!
!!! Otherwise PRIMARY could become SECONDARY (who knows why...)

```
(config)# mode multiple
(config)# failover lan interface MYFAILOVER ethernet2
(config)# failover lan enable
(config)# failover interface ip MYFAILOVER 172.16.1.1 255.255.255.0
standby 172.16.1.7
(config)# failover link MYFAILOVER ethernet2
(config)# failover lan key 1234567
(config)# failover lan unit secondary
(config)# failover
Detected an active mate...
```



- If the unit with a UR license in a failover pair fails and is removed from the configuration, the unit with the FO or FO_AA license will **not** automatically reboot every 24 hours; *it will operate uninterrupted until it is manually rebooted.*
- When the unit automatically reboots, the following message displays on the console:

```
=====NOTICE=====
This machine is running in secondary mode without
a connection to an active primary PIX. Please
check your connection to the primary system.
REBOOTING....
```

- ~~The ASA platform does not have this restriction.~~



- **Is sent in clear!**
 - ◆ Unless secured using a failover key (LAN-based FO only!)
- **Information communicated:**
 - ◆ The unit state (active or standby)
 - ◆ Power status
 - Cable-based failover only—available only on the Cisco PIX security appliance platform
 - ◆ Hello messages (keep-alives)
 - ◆ Network link status
 - ◆ MAC address exchange
 - ◆ **Configuration replication and synchronization**

Reason of the failover mac address



- Upon failover, the secondary unit becomes active and uses the IP and MAC addresses of the (failed) primary unit
- Now assume both FWs are turned off and the secondary boots first: It would use its burned-in MAC addresses
 - ◆ Since no contact to primary it becomes active immediately
- Later, when the primary is booted:
 - ◆ The secondary contacts the primary and uses the MAC address of the primary for its interfaces – this disrupts the traffic!
- But if you configure virtual MAC addresses for both the primary and the secondary then the secondary could immediately use the MAC of the primary when booted first



Intrusion Detection and Prevention



- **IDS on PIX (old)**
 - ◆ **Problem: no up-to-date signatures**
 - ◆ **Configuration: `ip audit` commands**



- IPS Software 5.1
- License required "only" to update signature files
- Use the **auto-upgrade-option enabled** command in the service host submode to configure automatic upgrades of signature files (*.pkg)

```
asa# session 1
(required to specify username/password upon first login)

AIP SSM# setup
(an interactive dialog starts to configure ip, telnet, ntp, banner,
hostname, etc.)
AIP SSM# reset
(reboot...)
AIP SSM# conf t
AIP SSM(config)# upgrade scp://1.2.3.4//upgrade/IPS-K9-maj-6.0-1-pkg
```



- **"Content Security and Control"**
 - ◆ Provides protection against viruses, spyware, spam, and other unwanted traffic
- **management port**
 - ◆ of the CSC SSM must be connected to your network to allow management of and automatic updates to the CSC SSM software. Additionally, the CSC SSM uses the management port for email notifications and syslogging.

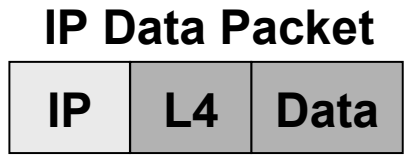


IPsec Repetition

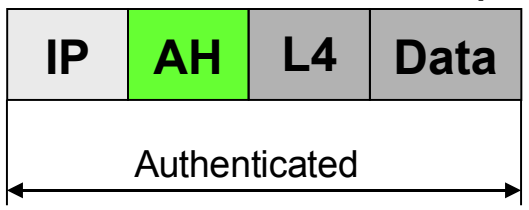


- **Two header choices:**
 - ◆ **AH** for message authentication only
 - ◆ **ESP** for message authentication and encryption
(=> you want this!)
- **Two modes:**
 - ◆ **Transport Mode** – if you want to have IPsec connections directly between your PCs or FWs
 - ◆ **Tunnel Mode** – if you want your *FWs* to encrypt and tunnel user's packets
 - The whole IP packet is encrypted and an additional outer IP-header is prepended (=> tunnel)
 - In a real network you want this!

IPsec Headers



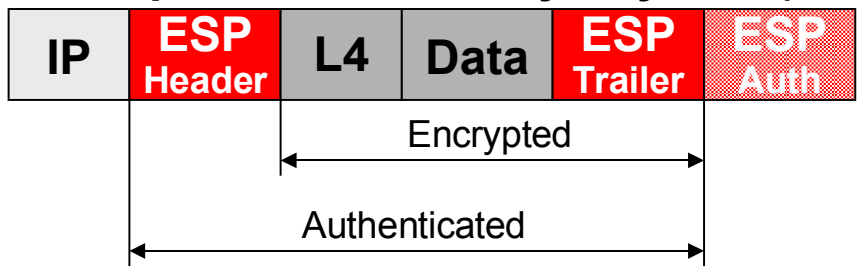
Authentication Header (AH)



Authentication only!
(must support keyed MD5)

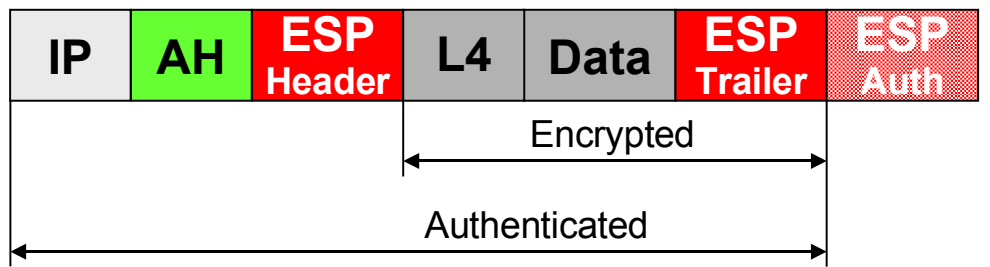
AND/OR

Encapsulation Security Payload (ESP)



Optional Encryption (DES CBC based) and optional authentication.

AH + ESP together:

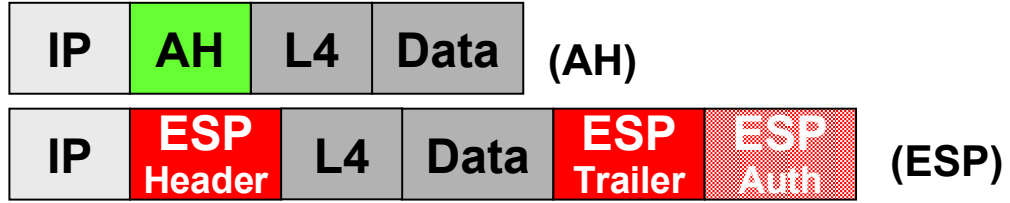


Modes

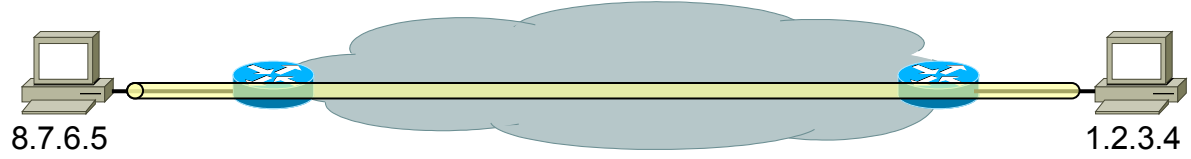


Transport Mode

- Only one IP header.
- Used for end-to-end sessions.
- Does not hide traffic patterns!



OR



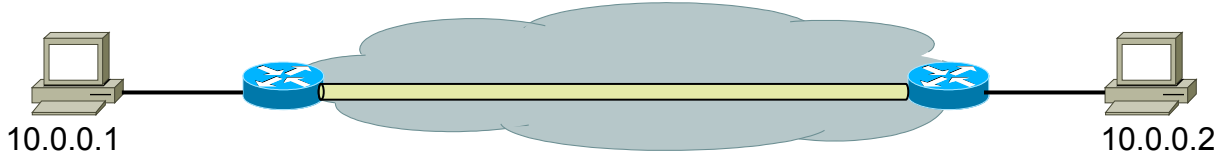
Tunnel Mode

- Whole original IP packet is IPsec-encapsulated.
- Used for VPNs.



Additional outer IP header

Original inner IP header maintains private addresses



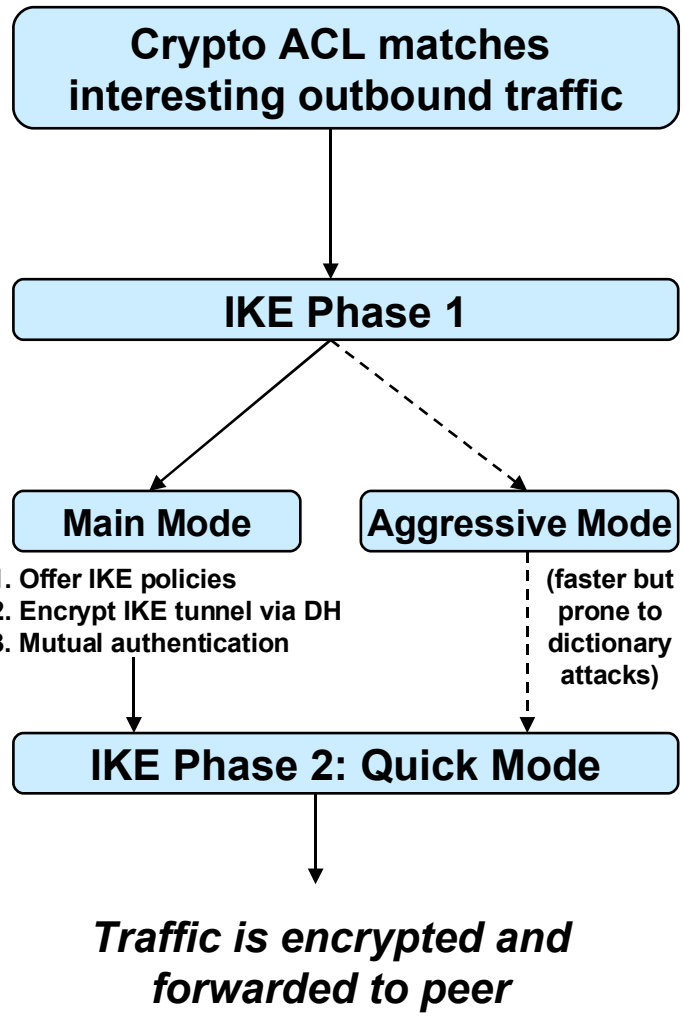


- **Data connections between two peers are known as Security Associations (SAs)**
 - ◆ SAs are unidirectional
 - ◆ That is, you need typically two of them
 - ◆ Multiple SA-pairs possible (e. g. with different cryptographic level for different data sessions)
- **IPsec requires to establish a key management SA**
 - ◆ Running the IKE protocol inside
 - ◆ This SA is bidirectional !

Procedures



For comparisons,
the router config:



IPsec tunnel must be established.

Negotiate IKE SA parameters and perform mutual auth.

Two choices. Prefer Main Mode which is more secure.

Negotiate IPsec SA parameters ("Transform Set")

```

! Must be an outbound ACL
(config)# access-list 101 permit tcp
10.10.10.0 0.0.0.255 10.20.20.0 0.0.0.255

! Enable IKE
(config)# crypto isakmp enable

! Specify pre-shared key used for peer
! authentication
(config)# crypto isakmp key MyKey address
172.30.2.2

! Create IKE policies
(config)# crypto isakmp policy 110
(config-isakmp)# authentication pre-share
(config-isakmp)# encryption des
(config-isakmp)# group 1
(config-isakmp)# hash md5
(config-isakmp)# lifetime 86400

! Configure (one or more) transform set(s)
(config)# crypto ipsec transform-set MY_SET
esp-3des esp-sha-hmac

! Bind anything in a crypto-map
(config)# crypto map MYMAP 120 ipsec-isakmp
(config-crypto-map)# match address 101
(config-crypto-map)# set peer 172.30.2.2
(config-crypto-map)# set transform-set MY_SET
(config-crypto-map)# set pfs group 1

! Attach crypto map to outgoing interface
(config-if)# crypto map MYMAP
  
```

Note (1)



- **Per default routers perform tunnel mode**
- **Default Phase 1 parameters are:**
 - ◆ Authentication: Certificates
 - ◆ Encryption: DES
 - ◆ MAC: SHA-1 MAC
 - ◆ DH: Group 1
- **IPsec does not support broadcasts or multicasts**
 - ◆ Problem with routing protocols
 - ◆ Use GRE instead (P2MP)
- **IKE uses UDP port 500**
- **Also manual SA establishment (without IKE) possible**
 - ◆ Requires manual configuration of session keys
- **SA Lifetime**
 - ◆ Default for IKE SA: 86400 sec (=1 day)
 - ◆ Default for IPsec SA: 3600 sec (=1 hour) OR 4 GByte data transfer
 - ◆ Keying material is discarded, IKE phase 2 (or also 1 if required) re-negotiation
 - New SA is negotiated before old SA expires (smooth transition)



- **When multiple crypto map entries?**
 - ◆ Different data flows to be handled with different security levels and/or peers
- **Map-number (=entry number)**
 - ◆ The lower, the higher the priority
 - ◆ At the crypto map set's interface, traffic is evaluated against higher priority map entries first
- **Perfect Forward Secrecy (PFS)**
 - ◆ New DH exchanged performed for each new phase 2 SA
 - ◆ Without PFS the same DH values of phase 1 are used



VPN Configuration



- Use `nat (inside) 0 <crypto-acl>` to avoid NAT on IPsec traffic
- Site-to-site VPN: Specify how to authenticate:
 - ◆ `(config) # isakmp identity`
`address | hostname | auto | key-id`
 - ◆ Used to identify the peer (IP address, group-name, certificate-DN-OU)

Tunnel Groups



- **tunnel-group <name> general-attr**
 - ◆ Specify general tunnel attributes here
 - ◆ E. g. for remote-access, specify address-pool and AAA server group for xauth
- **tunnel-group <name> ipsec-attr**
 - ◆ Specify pre-shared-key here
- **group-policy <name> attributes**
 - ◆ To specify additional group-specific attributes (pfs, dns, domain-name, ...)
 - ◆ Must be same name as tunnel-group
- **Optionally user specific attributes with the**
username xyz attributes command



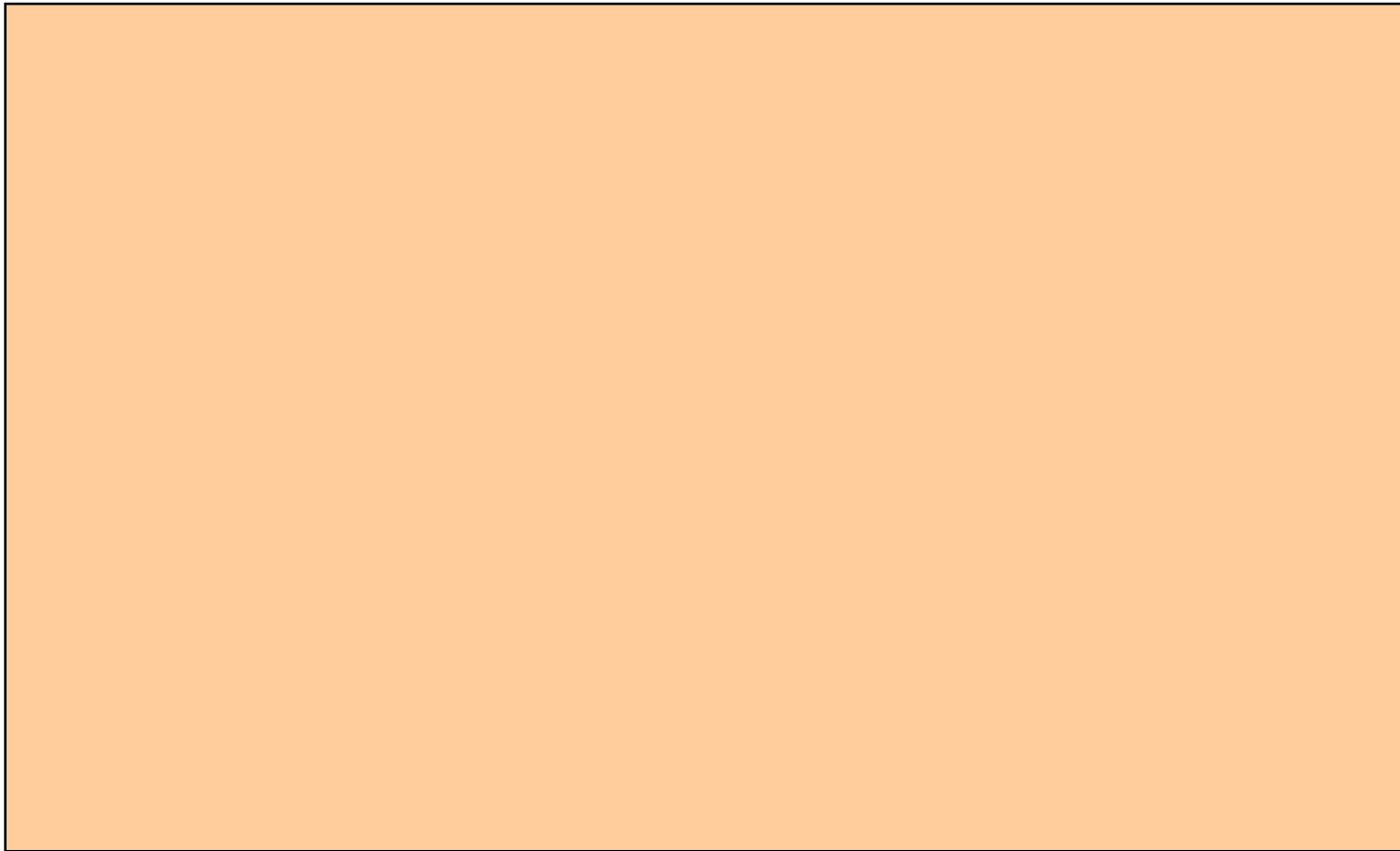
- **Needed to support tunnel mode on clients**
 - ◆ **Virtual tunnel interface created**
- **Available crypto map is augmented with dynamical crypto map**
 - ◆ Typically appended to be last one (no requirement but useful, otherwise perhaps problems with site-to-site VPNs)
- **Easy VPN remote supports two modes:**
 - ◆ **Client mode**
 - Automatic NAT/PAT translation
 - ◆ **Network extension mode**
 - No PAT—assumes clients have fully routable addresses

Easy VPN Server Configuration



- 1. Create ISAKMP policy for remote VPN Client access**
- 2. Create IP address pool**
- 3. Define group policy for mode configuration push**
- 4. Create transform set**
- 5. Create dynamic crypto map**
 - ◆ Better don't use same name as static crypto map
- 6. Assign dynamic crypto map to static crypto map**
- 7. Apply crypto map to PIX Firewall interface**
- 8. Configure XAUTH**
 - ◆ Either against TACACS+/RADIUS server or local
 - ◆ Append keyword no-xauth at site-to-site policies (password config)
 - Otherwise user credentials are requested
 - ◆ Username/password is not bound to a group
 - Lock group feature (only on IOS and PIX OS 7.0) => "user@group" requested
- 9. Configure NAT and NAT 0**
 - Typically no NAT for outbound traffic into IPsec tunnels (to VPN clients)
- 10. Enable IKE DPD**

Config Example



2) Dynamic Crypto Map



```
! IKE Phase 2: Create transform set for IPsec SAs:  
(config)# crypto ipsec transform-set RMTUSER1 esp-des esp-sha-hmac  
  
! Create dynamic crypto map:  
(config)# crypto dynamic-map DYNAMAP 10 set transform-set RMTUSER1  
  
! Assign dynamic crypto map to static crypto map:  
(config)# crypto map STATMAP 1000 ipsec-isakmp dynamic DYNAMAP  
  
! Assign crypto map to outside interface:  
(config)# crypto map STATMAP outside
```

3) XAUTH



```
! Enable AAA login authentication, server IP address and shared key:
(config)# aaa-server MYTACACS protocol tacacs+
(config)# aaa-server MYTACACS (inside) host 10.0.0.15 cisco123 timeout 5

! Enable XAUTH for the crypto map:
(config)# crypto map STATMAP client authentication MYTACACS

! Configure NAT and NAT 0 to disable NAT for IPsec packets:
(config)# access-list 101 permit ip 10.0.0.0 255.255.255.0
        10.0.11.0 255.255.255.0
(config)# nat (inside) 0 access-list 101
(config)# nat (inside) 1 0.0.0.0 0.0.0.0 0 0
(config)# global (outside) 1 interface

! Enable DPD:
! AKA "paranoid keepalives"
(config)# isakmp keepalive 30 10
```

- **Note: If there is ALSO a site-to-site VPN, append the keyword no-xauth at the site-to-site policies (password config)**
 - ◆ Router cannot provide any XAUTH credentials

PIX as Remote (only small PIXes)



- **Configure:**
 - ◆ Group
 - ◆ Password (XAUTH password can also be specified, if needed)
 - ◆ VPN server address
- **Modes**
 - ◆ Client mode (PIX simulates normal VPN client: receives IP address from server and performs PAT with this assigned address)
 - ◆ Network extension mode (simulates site-to-site VPN with dynamic address – client addresses visible from central site)

```
(config)# vpngroup training password cisco123
(config)# vpnclient username student1 password training
(config)# vpnclient server 192.168.1.2
(config)# vpnclient mode network-extension-mode
(config)# vpnclient enable
```



- **Allow encrypted traffic to enter and leave the same interface:**

```
! Note: with version 7.2 and later, this command allows any traffic  
!       to enter and leave the same interface!  
(config)# same-security-traffic permit intra-interface
```

WebVPN



- **Java Applet is a local TCP-Server-Application which listens to incoming connections**
 - ◆ **Either on high port numbers to the loopback (E.g. 127.0.0.1:62231)**
 - ◆ **Or by accessing hostnames which had been entered to local hosts file, e. g.**
 - 127.0.0.3 application1
 - 127.0.0.4 application2
- **Upon connections “to the Java-Applet” the packets are forwarded into SSL tunnels**
 - ◆ **Dedicated additional tunnels!**



- **PIX/ASA allows to turn on “proxy servers”**
 - ◆ **Imap4s ... accepts incoming IMAP4 over SSL (Port 993)**
 - ◆ **Pop3s ... accepts incoming POP3 over SSL (Port 995)**
 - ◆ **Smtps ... accepts incoming SMTP over SSL (Port 988)**

- **These incoming sessions are**
 - ◆ **Locally terminated by the ASA**
 - ◆ **Then decrypted and forwarded to a configured server**



- **MAPI**
 - ◆ **Messaging Application Interface**
 - ◆ **Developed by MS to allow programs to utilize E-Mail functions**

- **CIFS**
 - ◆ **Common Internet File System**
 - ◆ **Developed by MS**



■ **Disadvantages**

- ◆ **Cannot prevent split tunneling - client can always access anything**
- ◆ **Browser caches credentials**
- ◆ **Only for RA-VPN**
- ◆ **TCP-based attacks, SQNRs desync**

Transparent Firewall



- FW makes transparent bridging
- **Only two interfaces!**
 - ◆ Different VLANs but same IP networks !
 - ◆ Per context! Each context must use different interfaces !!!
- Also for multiple context mode
 - ◆ But then all contexts in transparent mode
- **By default only ARP may pass through**
 - ◆ Other traffic must be permitted by ACL (normal or EtherType based)
 - ◆ EtherType-ACLs can only match Ethernet-2 and SNAP encapsulated packets
- **Not supported:**
 - ◆ NAT, Routing Protocols, IPv6, DHCP relay, QoS, Multicast, VPN termination for through traffic



- **[no] firewall transparent**
 - ◆ Clears configuration!
 - ◆ Verify with `show firewall`
- **Optionally mitigate MITM via ARP inspection**
 - ◆ First configure a static IP-MAC table:
`arp outside 5.4.3.2 000c.abcd.9876`
 - ◆ Then enable ARP inspection and optionally specify what to do if ARP entries could not be found in the static table:
`arp inspection outside enable [flood | no-flood]`

FW is not really a bridge!



- **show mac-address-table**
 - ◆ Similar as bridge, source MAC addresses are entered in the table (“learning”)
- **But if destination MAC not found there is no “flooding” !!!**
 1. The packet is dropped
 2. If destination local, an ARP is sent to learn the associated port
 3. If destination not local, a PING is sent to learn the associated port
- ◆ **Recommendations:**
 - ◆ Add static entries:
`mac-address-table static inside 000c.affe.affe`
 - ◆ Disable learning e. g. on outside interface: `mac-learn outside disable`



- **ACL Types**
 - ◆ Extended (normal)
 - ◆ Ethertype – only Type field and BPDU-frames
- **# arp <interface> <ip> <mac>**
 - ◆ static interface to mac to ip assignment
- **# arp-inspection <interface> enable [flood | no-flood]**
 - ◆ Every ARP packet is checked against static table
 - ◆ Malicious **ARP-response** is dropped - Prevents ARP spoofing!
 - ◆ **flood**: if IP not known than **ARP-request** is flooded through all interfaces

AIP-SSM



- **Internal**
 - ◆ **Gigabit-Ethernet as main data path to ASA for both inline and promiscuous IPS packets**
 - ◆ **Plus 10/100 Ethernet as control channel**
- **External**
 - ◆ **10/100/1000 Ethernet for downloading AIP-SSM software and access**
- **Roles**
 - ◆ **Administrator – highest privilege**
 - ◆ **Operator – tune signatures, manage routers**
 - ◆ **Viewers**
 - ◆ **Service – direct access to bash shell (support and troubleshooting)**

Facts



Commands



```
!!! Load recovery image from tftp. First define network details:
ASA# hw module 1 recover configure
!!! Then start download:
ASA# debug module !!! observe anything
ASA# hw module 1 recover boot

ASA# show module 1 [detail] !!! VERIFY IF STATUS IS UP !!!

ASA# session 1
...enter cisco/cisco as default password...

Sensor# setup
```



This and That



- **Console idle timeout:**

```
! In seconds, 0 means infinity  
(config)# console timeout 300
```

- **Configure login authentication for console:**

```
(config)# username max password moritz  
(config)# aaa authentication serial console LOCAL
```

SSH and Telnet



Allow Telnet to the PIX:

```
! First configure a telnet password
(config)# passwd sanfran
! Specify allowed hosts
(config)# telnet 10.1.1.30 255.255.255.255 inside
```

Simple telnet config for one host, using single password only

```
(config)# username cisco password cisco
(config)# aaa authentication telnet console LOCAL
```

Or use usernames and passwords

Allow SSH to the PIX:

```
(config)# username herbert password cisco
(config)# aaa authentication ssh console LOCAL
(config)# hostname mypix
(config)# domain-name xyz.com
(config)# crypto key generate rsa modulus 2048
...
(config)# write mem !!! Store keys in flash
! Specify allowed hosts:
(config)# ssh 10.1.1.30 255.255.255.255 inside
! Set inactivity timeout in minutes:
(config)# ssh timeout 30
(config)# ssh version 2
```

Basic SSH configuration with usernames and passwords.



- **Every subinterface requires a VLAN ID to pass traffic**
 - ◆ Also the `nameif` command is required
- **The main interface forwards untagged traffic (native VLAN)**
 - ◆ **Omit the `nameif` command if the main interface should not pass any traffic**

PING does not work?



(or other unreachability issues)

- Don't forget to permit echo-reply on outside interface, otherwise no ping from inside hosts possible
 - ◆ There is no ICMP inspection by default!
- If an ACL has been removed (e.g. via clear config access-list ...) and reconfigured then the access-group must be entered again !!!
- Does the Windows XP Firewall block?
- Check the routing table!
 - ◆ Also on hosts:
Is there also a default-gw configured?



- **PIX Device Manager (PDM)**
 - ◆ Requires versions ≤ 6.3
 - ◆ Supports FWSM
- **Adaptive Security Device Manager (ASDM)**
 - ◆ Requires versions ≥ 7.0
 - ◆ Supports PIX and ASA
 - ◆ Supports FWSM ≥ 3.1

The Eight Syslog Levels



- **0 Emergencies**
- **1 Alerts**
- **2 Critical**
- **3 Errors**
- **4 Warnings**
- **5 Notifications**
- **6 Info**
- **7 Debugging**



- **Show ip address** shows two sections:
 - ◆ **System IP address**: in FO configuration the IP address of the active system
 - ◆ **Current IP address**: in FO configuration
 - On the standby device: the standby IP address
 - On the active device: same as system IP address
 - ◆ In non-FO configurations
- **clear local-host**
 - ◆ Clears xlate and connections