TOMORROW starts here.

cisco

Cisco live!

Troubleshooting ASA Firewalls

BRKSEC-3020

Andrew Ossipov
Technical Marketing Engineer

Cisco live!

Your Speaker

Andrew Ossipov

aeo@cisco.com

Technical Marketing Engineer

8+ years in Cisco TAC

16+ years in Networking

CISCO CERTIFIED CCDP DESIGN

CISCO CERTIFIED CCIE

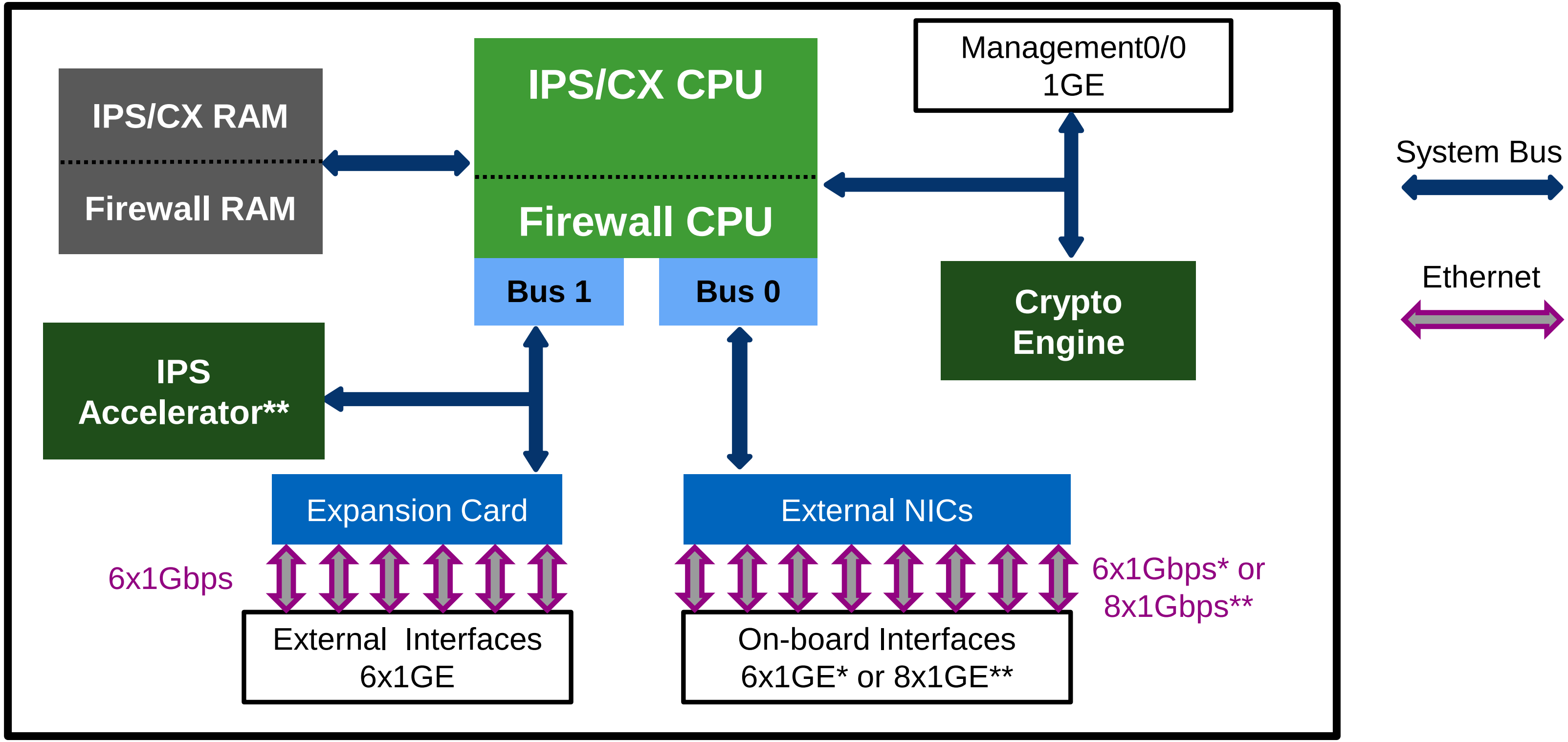
CISSP

Agenda

- ASA Architecture
- Packet Flow
- Diagnostic Messages and Outputs
- Troubleshooting Tools
- Case Studies
- Best Practices

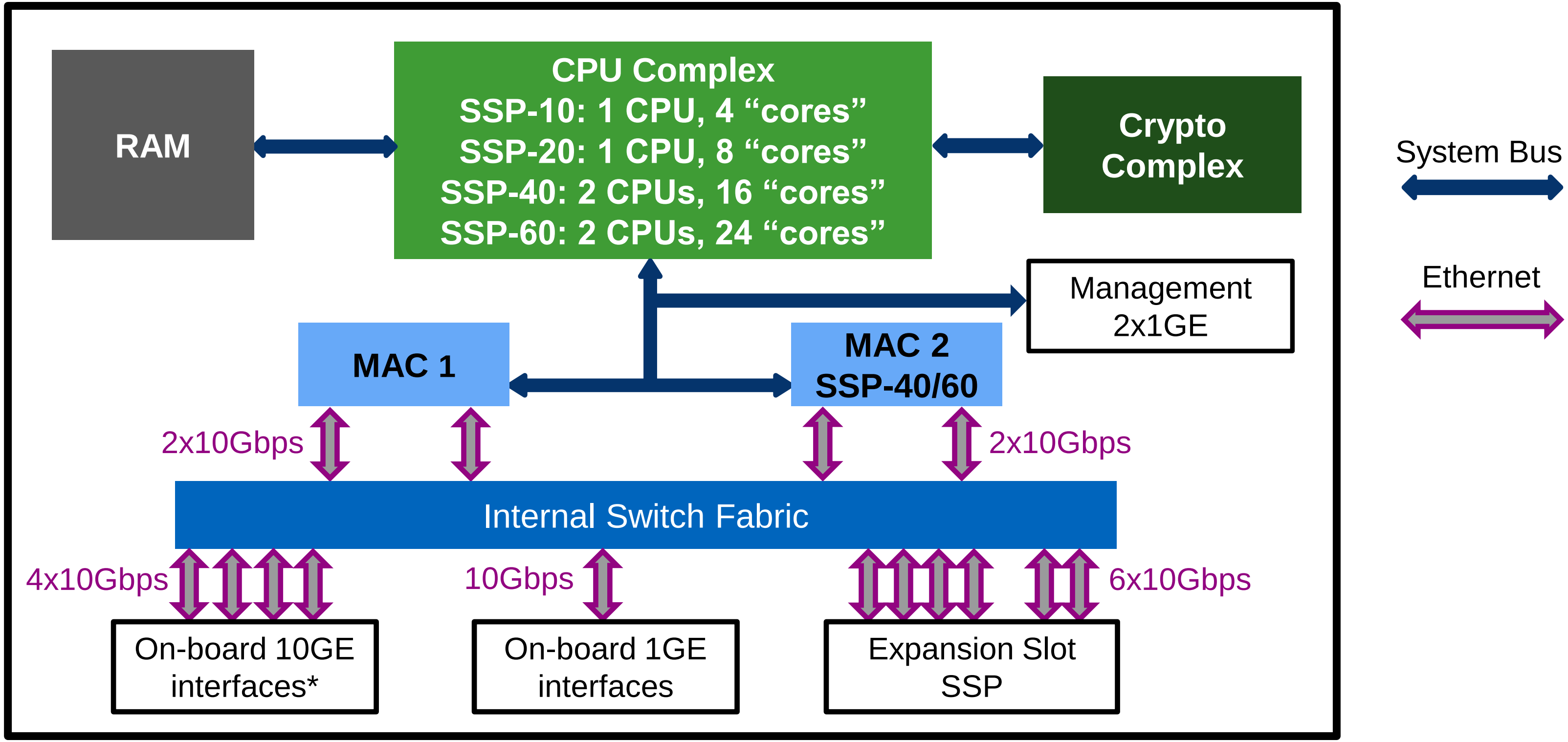
ASA Architecture

ASA 5500-X Block Diagram



*ASA5512-X and ASA5515-X
** ASA5525-X and higher

ASA 5585-X Block Diagram

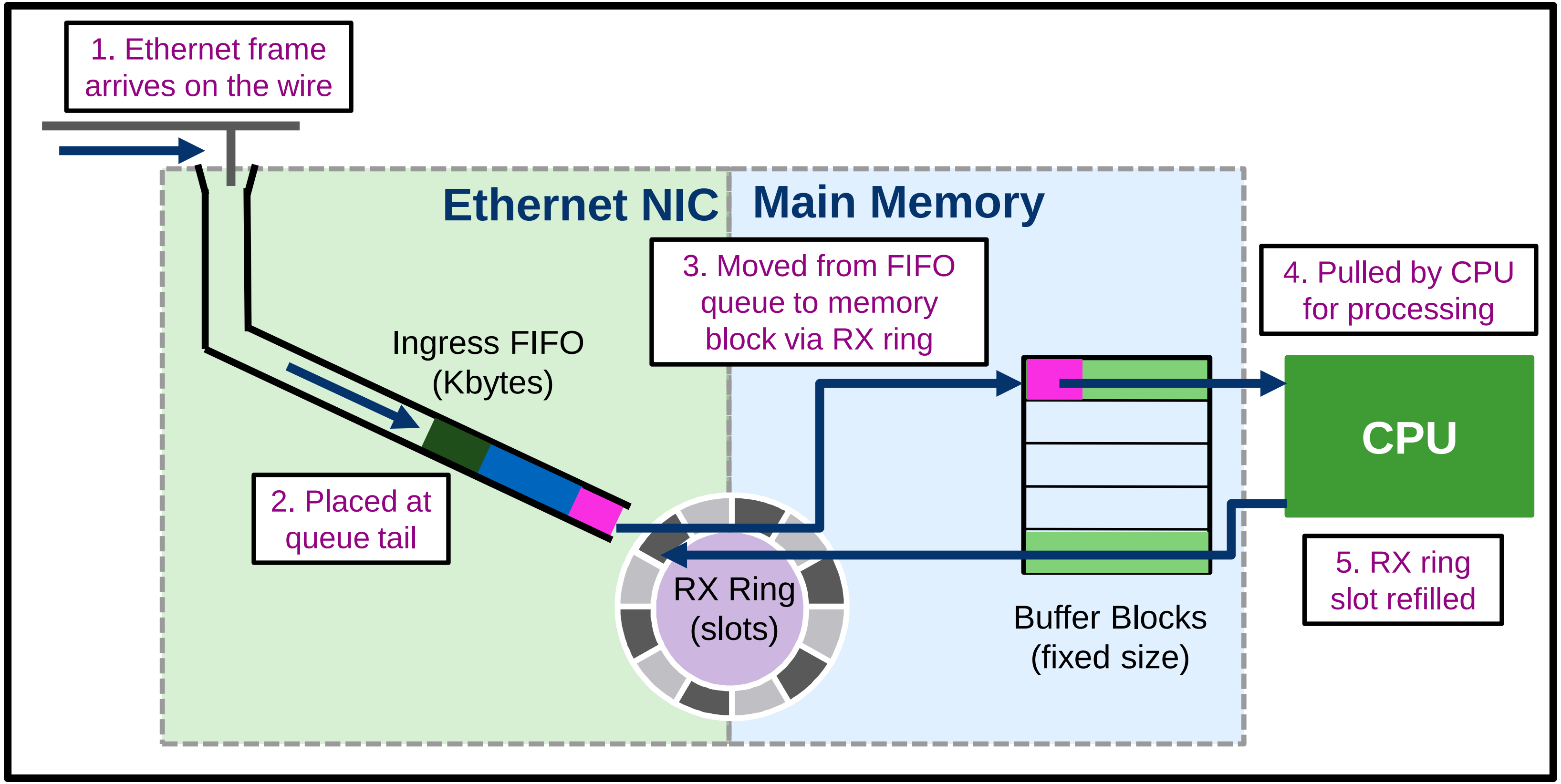


*2 on SSP-10/20 and 4 on SSP-40/60

Ingress Frame Processing

- Frames are received from wire into ingress FIFO queues
  - 32/48KB on 1GE (except management ports), 512KB on 10GE
- Network Interface Controller (NIC) moves frames to main memory via RX rings
  - Each ring slot points to a main memory address (“block” or “buffer”)
  - Single RX ring per 1GE, multiple RX rings per 10GE
  - Shared RX rings on 10GE MACs (ASA5585/SM) and 1GE uplink (ASA5505)
- CPU periodically “walks” through all RX rings
  - Pull new ingress packet blocks for processing
  - Refill slots with pointers to other free blocks

NIC Architecture



1. Ethernet frame arrives on the wire

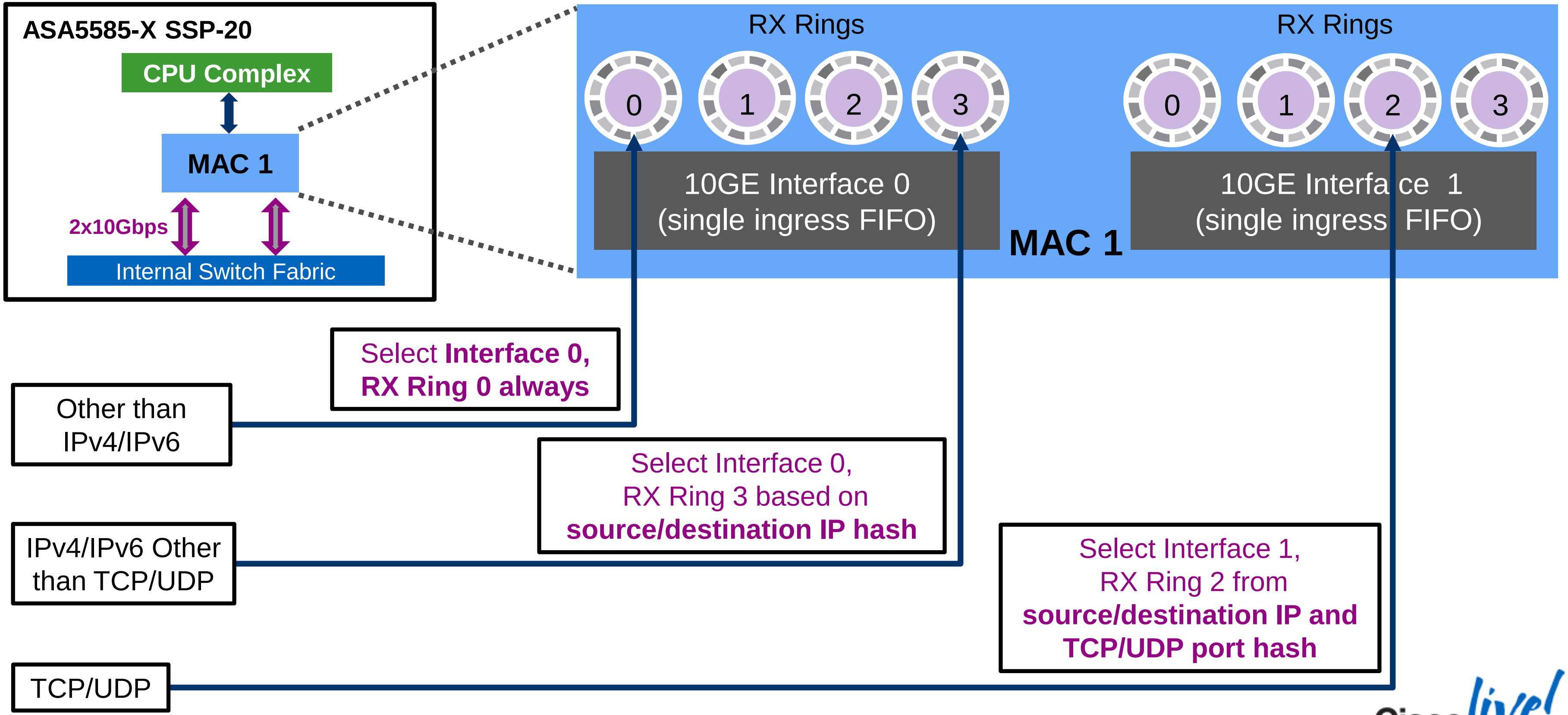
2. Placed at queue tail

3. Moved from FIFO queue to memory block via RX ring

4. Pulled by CPU for processing

5. RX ring slot refilled

Ingress Load-Balancing on 10GE and MAC



ASA5585-X SSP-20

CPU Complex

MAC 1

2x10Gbps

Internal Switch Fabric

RX Rings

0 1 2 3

10GE Interface 0
(single ingress FIFO)

MAC 1

RX Rings

0 1 2 3

10GE Interfa ce 1
(single ingress FIFO)

Other than IPv4/IPv6

Select Interface 0, **RX Ring 0 always**

IPv4/IPv6 Other than TCP/UDP

Select Interface 0, RX Ring 3 based on **source/destination IP hash**

TCP/UDP

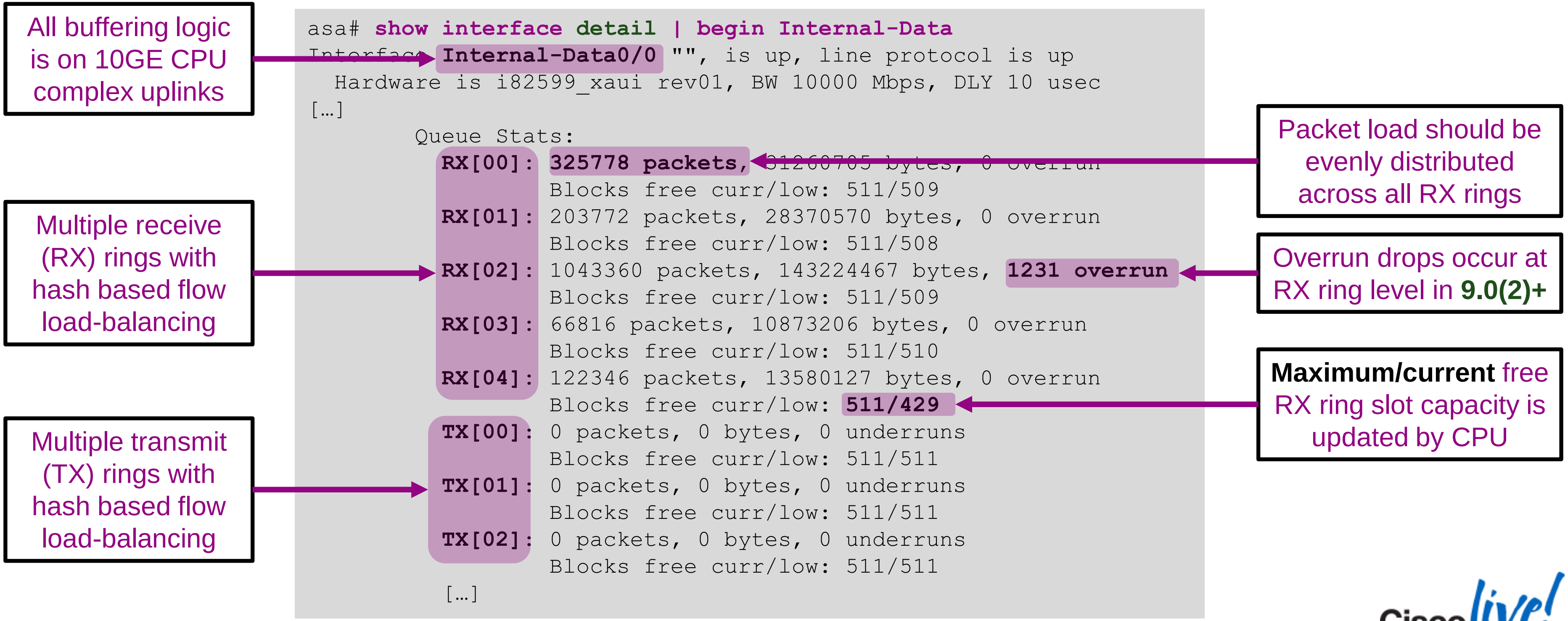
Select Interface 1, RX Ring 2 from **source/destination IP and TCP/UDP port hash**

NIC Performance Considerations

- If ingress FIFO is full, frames are dropped
  - No free slots in RX ring (CPU/memory bound)
  - **No buffer** on memory move errors, **overrun** on FIFO drops
- FIFO is not affected by packet rates, but RX rings are
  - Fixed memory block size regardless of actual frame size
  - Ingress packet bursts may cause congestion even at low bits/sec
- Maximise frame size and minimise rate for best efficiency
  - Jumbo frames supported on ASA5500-X, ASA5580, ASA5585-X, and ASASM
  - Configure **jumbo-frame reservation**, reload, and raise the interface MTU
  - Do not forget **sysopt connection tcpmss 0**

10GE MAC Interface Information

- Check Internal-Data 10GE MAC interfaces on ASA5585 and ASASM for errors



```
asa# show interface detail | begin Internal-Data
Interface Internal-Data0/0 "", is up, line protocol is up
  Hardware is i82599_xaui rev01, BW 10000 Mbps, DLY 10 usec
[…]
        Queue Stats:
          RX[00]: 325778 packets, 31260705 bytes, 0 overrun
                  Blocks free curr/low: 511/509
          RX[01]: 203772 packets, 28370570 bytes, 0 overrun
                  Blocks free curr/low: 511/508
          RX[02]: 1043360 packets, 143224467 bytes, 1231 overrun
                  Blocks free curr/low: 511/509
          RX[03]: 66816 packets, 10873206 bytes, 0 overrun
                  Blocks free curr/low: 511/510
          RX[04]: 122346 packets, 13580127 bytes, 0 overrun
                  Blocks free curr/low: 511/429
          TX[00]: 0 packets, 0 bytes, 0 underruns
                  Blocks free curr/low: 511/511
          TX[01]: 0 packets, 0 bytes, 0 underruns
                  Blocks free curr/low: 511/511
          TX[02]: 0 packets, 0 bytes, 0 underruns
                  Blocks free curr/low: 511/511
          […]
```

All buffering logic is on 10GE CPU complex uplinks

Multiple receive (RX) rings with hash based flow load-balancing

Multiple transmit (TX) rings with hash based flow load-balancing

Packet load should be evenly distributed across all RX rings

Overrun drops occur at RX ring level in **9.0(2)+**

Maximum/current free RX ring slot capacity is updated by CPU

CPU Packet Processing

- NIC moves packets from Ethernet to memory
- All packets are processed by the CPU complex in software
- Data Path CPU process checks all inbound packets **sequentially**
  - Stateful checks are applied to every single packet
  - Fastpath, Slowpath, Control Plane
- New connection requests are directed to **Slowpath**
  - Access Control List check, NAT xlate creation, conn creation, logging
- Existing connections are processed in **Fastpath**
  - Bypass ACL check, find egress interface, apply NAT, transmit packet
- **Control Plane** performs Application Inspection and management

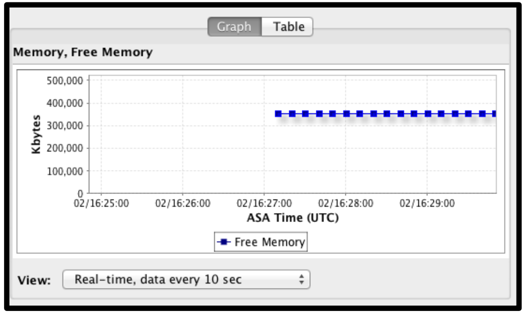
Multiple-Core Platforms

- Some firewalls have more than one CPU “cores”
  - ASA5500-X, ASA5580, ASA5585-X, ASASM
- Multiple-core ASAs run many Data Path processes in parallel
  - Only one core can “touch” a single connection at any given time
- One core runs Control Path process at all times
  - Dedicated Control Plane process that is separate from Data Path
  - System-wide tasks and everything that cannot be accelerated in Data Path

ASA Memory

- ASA memory is used by configuration, processes, transit packets

```
asa# show memory
Free memory:              250170904 bytes (47%)
Used memory:              286700008 bytes (53%)
-------------        ------------------
Total memory:             536870912 bytes (100%)
```



- If available memory trends down over time, call Cisco TAC

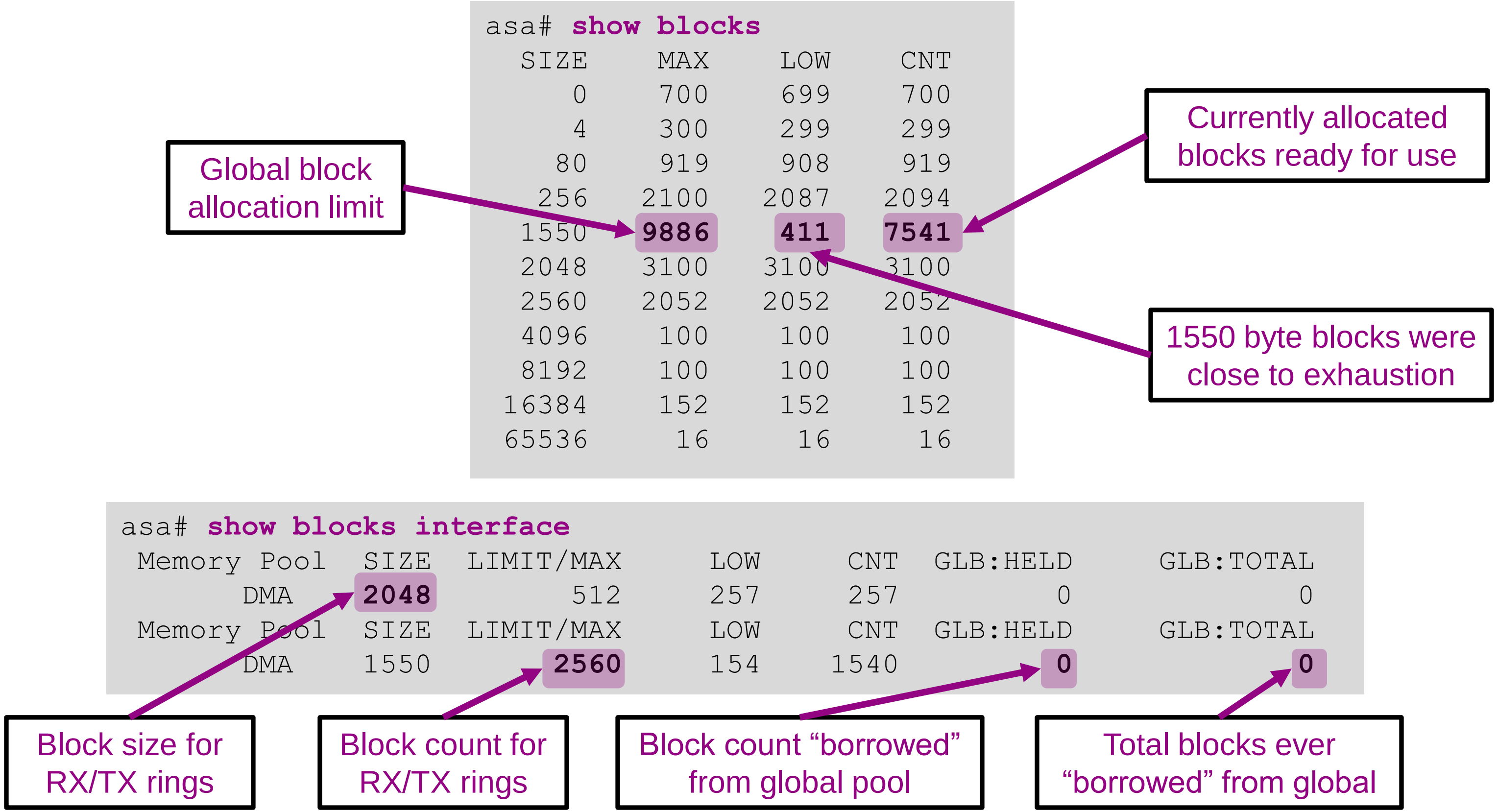
```
%ASA-3-211001: Memory allocation Error
```

  – CISCO-ENHANCED-MEMPOOL-MIB.my for accurate SNMP counters in **ASA 8.4+**
  – Free memory may not recover immediately after conn spike due to cashing

- Memory block depletion leads to packet drops and instability

```
%ASA-3-321007: System is low on free memory blocks of size 1550 (10 CNT out of 7196 MAX)
```

Memory Blocks on ASA



```
asa# show blocks
  SIZE    MAX    LOW    CNT
     0    700    699    700
     4    300    299    299
    80    919    908    919
   256   2100   2087   2094
  1550   9886    411   7541
  2048   3100   3100   3100
  2560   2052   2052   2052
  4096    100    100    100
  8192    100    100    100
 16384    152    152    152
 65536     16     16     16
```

Global block allocation limit

Currently allocated blocks ready for use

1550 byte blocks were close to exhaustion

```
asa# show blocks interface
 Memory Pool  SIZE  LIMIT/MAX     LOW     CNT  GLB:HELD     GLB:TOTAL
       DMA    2048        512     257     257         0             0
 Memory Pool  SIZE  LIMIT/MAX     LOW     CNT  GLB:HELD     GLB:TOTAL
       DMA    1550       2560     154    1540         0             0
```

Block size for RX/TX rings

Block count for RX/TX rings

Block count “borrowed” from global pool

Total blocks ever “borrowed” from global

Maximum ACL Limits

- ACL table size is only bound by available memory
- Compiled into binary structure, no performance advantage from order
- Each ACE uses a minimum of 212 bytes of RAM
- Connection rate is impacted beyond maximum recommended values

	5510	5520	5540	5550	5580-20	5580-40
Maximum recommended	80K	200K	375K	550K	1M	2M

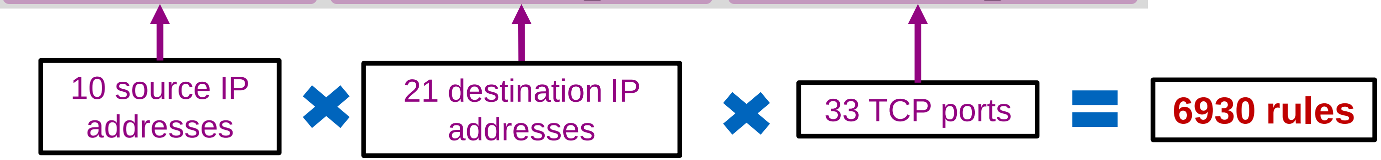
	5505	5512-X	5515-X	5525-X	5545-X	5555-X	5585-10	5585-20	5585-40	5585-60	ASASM
Maximum recommended (8.4+)	25K	100K	100K	250K	400K	600K	500K	750K	1M	2M	2M

- Issue **show access-list | include elements** to see how many ACEs you have

ACE Explosion with Object Groups

- All configured ACLs are expanded before programming

```
access-list IN permit tcp object-group INSIDE object-group DMZ_SERVERS object-group TCP_SERVICES
```



10 source IP addresses × 21 destination IP addresses × 33 TCP ports = **6930 rules**

- Nested Object Groups magnify the impact
  - Add a new source Object Group with 25 additional objects
  - Result: (10+25) x 21 x 33 = **24,255 rules** (ACEs)
- ACL Optimisation prevents the Object Group expansion
  - Significant reduction in memory utilisation, not so much on CPU

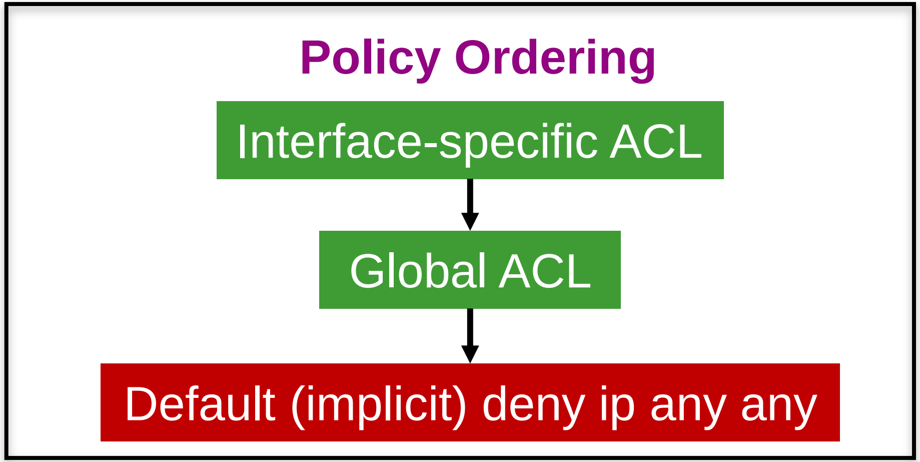
```
asa(config)# object-group-search access-control
```

- Cisco Security Manager (CSM) offers many ACL optimisation tools

Global ACLs

- Available in **ASA 8.3+**
- Apply the same security policy inbound to all interfaces
  - Useful for migrations from some vendors

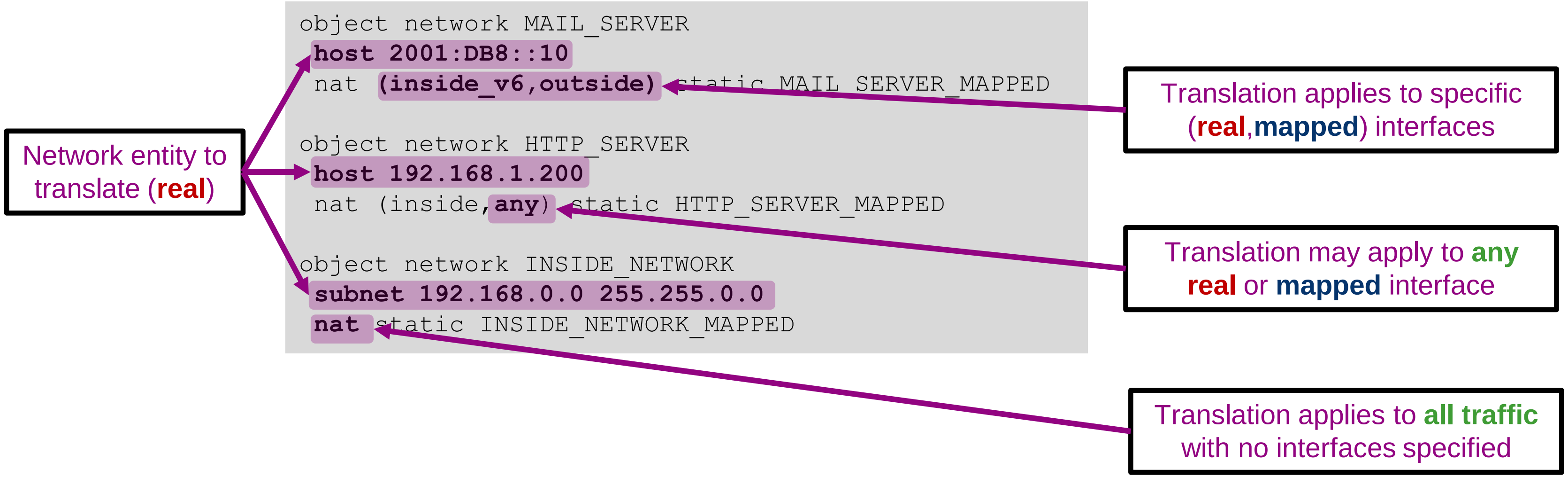
```
asa(config)# access-group <access_list> global
```



Network Object NAT

For your reference

- Simplest form of defining translation policy for Unified Objects
  - Only **one** translation rule per object
  - Configured network IP is **real**, translated is **mapped**
  - Applies to **all** traffic to or from the object, use interfaces names to limit scope



```
object network MAIL_SERVER
 host 2001:DB8::10
 nat (inside_v6,outside) static MAIL_SERVER_MAPPED

object network HTTP_SERVER
 host 192.168.1.200
 nat (inside,any) static HTTP_SERVER_MAPPED

object network INSIDE_NETWORK
 subnet 192.168.0.0 255.255.0.0
 nat static INSIDE_NETWORK_MAPPED
```

Network entity to translate (**real**)

Translation applies to specific (**real**,**mapped**) interfaces

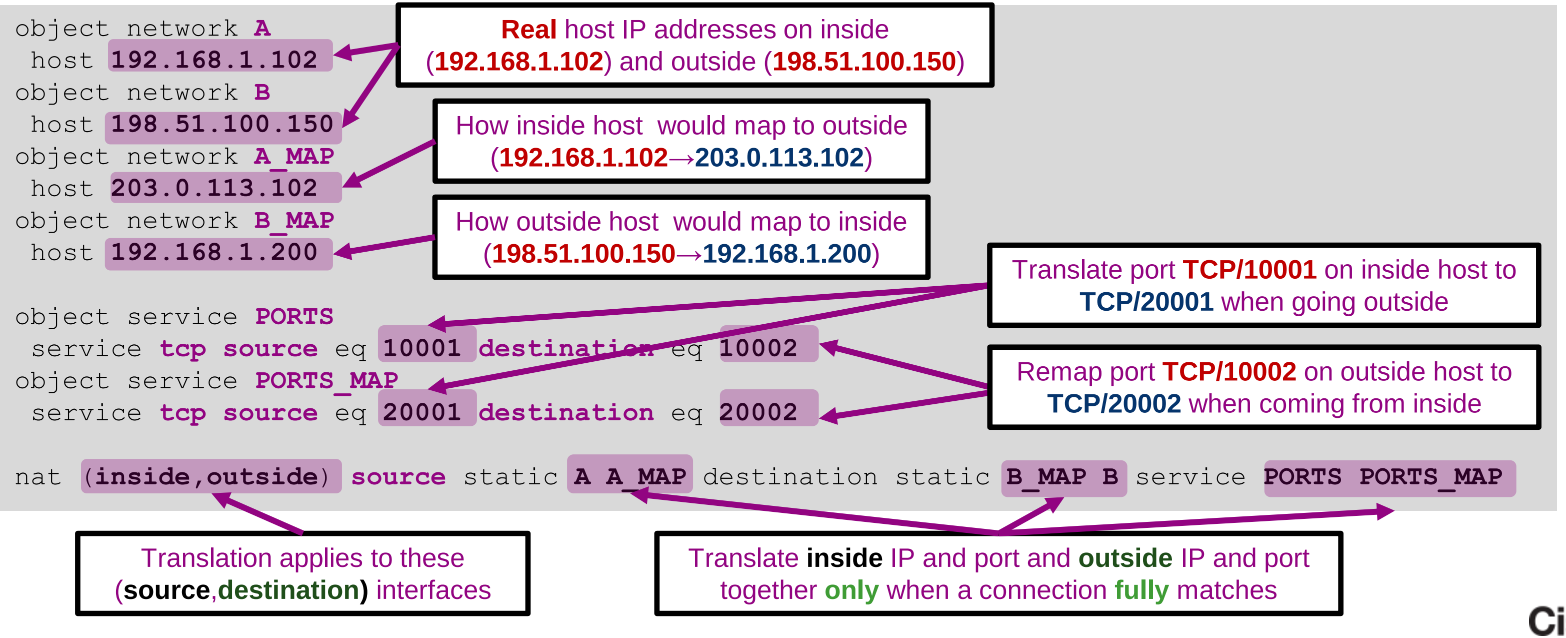
Translation may apply to **any** **real** or **mapped** interface

Translation applies to **all traffic** with no interfaces specified

Twice NAT

For your reference

- Match and translate packets on source and destination **together**
  - Similar to Network Object NAT, but cannot use in-line IP
  - A dynamic translation can **only** pair with a static one



```
object network A
 host 192.168.1.102
object network B
 host 198.51.100.150
object network A_MAP
 host 203.0.113.102
object network B_MAP
 host 192.168.1.200

object service PORTS
 service tcp source eq 10001 destination eq 10002
object service PORTS_MAP
 service tcp source eq 20001 destination eq 20002

nat (inside,outside) source static A A_MAP destination static B_MAP B service PORTS PORTS_MAP
```

Real host IP addresses on inside (**192.168.1.102**) and outside (**198.51.100.150**)

How inside host would map to outside (**192.168.1.102→203.0.113.102**)

How outside host would map to inside (**198.51.100.150→192.168.1.200**)

Translate port **TCP/10001** on inside host to **TCP/20001** when going outside

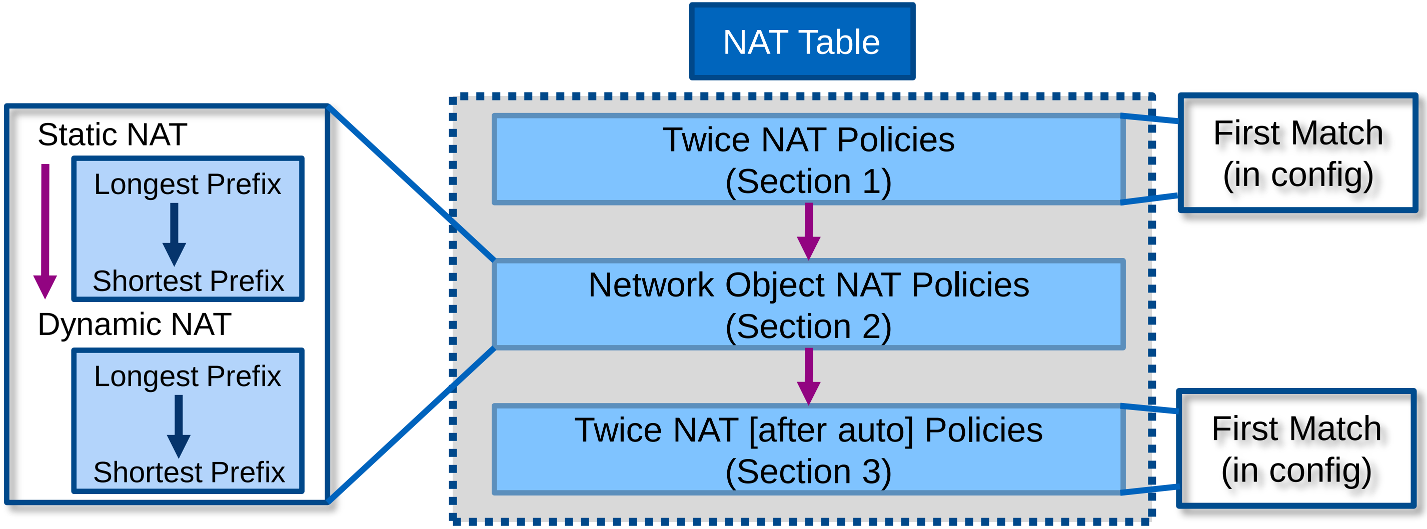
Remap port **TCP/10002** on outside host to **TCP/20002** when coming from inside

Translation applies to these (**source**,**destination**) interfaces

Translate **inside** IP and port and **outside** IP and port together **only** when a connection **fully** matches

NAT Order of Operation In ASA 8.3+

- The ASA configuration is compiled into the NAT table
  - Twice NAT rules always match and translate both source and destination
  - Network object NAT translates destination **first**, then source (separate rules)
- The **show nat** command will display the NAT table in order



NAT Traffic Diversion

- Network Object and Twice NAT override routing table on inbound
  - Network Object NAT **diverts** packets to real interface **only** for actual translation

```
object network DMZ_FTP
 host 198.51.100.200
 nat (dmz,outside) static 198.51.100.200
object network DMZ_MAIL
 host 172.16.171.125
 nat (dmz,inside) static 192.168.1.201
```

Identity translation, so inbound packets from **outside** to **198.51.100.200** use routing table

Actual translation happens, so inbound packets from **inside** to **192.168.1.201** will always divert to **172.16.171.125** on **DMZ**

  - Twice NAT rules divert packets to respective interfaces by default

Traffic from **192.168.2.0** on **outside** to **192.168.1.0** is diverted to **inside**

Traffic from **192.168.1.0** on **inside** to **192.168.2.0** is diverted to **outside**

```
nat (inside,outside) source static 192_168_1_0 192_168_1_0 destination static 192_168_2_0 192_168_2_0
```

- Best to disable divert for broad identity Twice NAT rules

```
nat (inside,any) source static 10_0_0_0 10_0_0_0 destination static 10_0_0_0 10_0_0_0 route-lookup
```

All traffic to **10.0.0.0/8** would be diverted to **inside**

Force **routing table** lookup to prevent problems

Real IP ACLs

- Finally, a reminder that **ASA 8.3+** uses **real** IP addresses in ACL
  - Pre-NAT for source and post-NAT for destination IP addresses

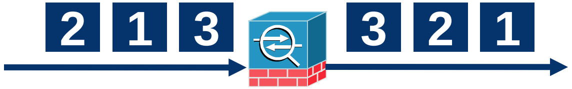
```
object network obj-WebServer
 host 10.3.19.50
 nat (inside,outside) static 198.51.100.50
!
access-list allowIn permit tcp any host 10.3.19.50 eq 80
!
access-group allowIn in interface outside
```

Application Inspection Engines

- Primarily perform embedded IP rewrites and open ACL pinholes
  - Very few engines enforce protocol compliance
  - Inspection Policy Maps can be used to match protocol fields for custom actions

```
policy-map global_policy
 class inspection_default
  inspect ftp FTP_BLOCK_PUT_COMMAND
```

  - **Exclusive** matching, but class **inspection_default** allows multiple **inspect** actions
- Very heavy performance impact on ASA due to extra work
  - Application inspection typically happens in Control Path (single core)
  - TCP traffic has to be put in the correct order first



Packet Flow

Understanding Packet Flow

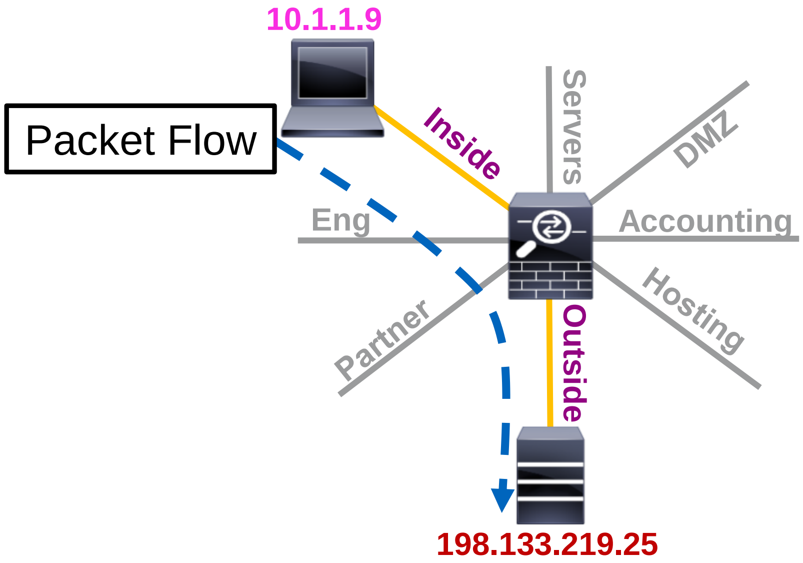
- To effectively troubleshoot a connectivity problem, one must first understand the packet path through the network
- Attempt to isolate the problem down to a single device
- Then perform a systematic walk of the packet path through the device to determine where the problem could be
- For problems relating to the Cisco ASA, always
  - Determine the flow: Protocol, Source IP, Destination IP, Source Port, Destination Port
  - Determine the logical (named) interfaces through which the flow passes

```
TCP outside  172.16.164.216:5620 inside  192.168.1.150:50141, idle 0:00:00, bytes 0, flags saA
```

All firewall connectivity issues can be simplified to two interfaces (ingress and egress) and the policies tied to both

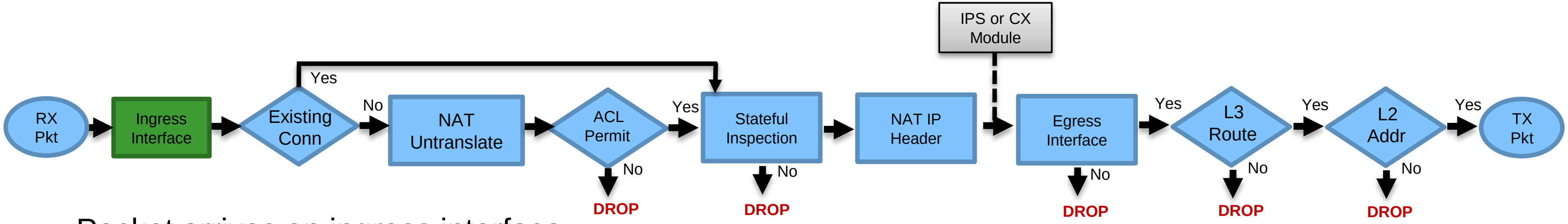
Example Flow

- TCP Flow
  - Source IP : 10.1.1.9 Source Port : 11030
  - Destination IP : 198.133.219.25 Destination Port : 80
- Interfaces
  - Source: **Inside** Destination: **Outside**



With the Flow defined, examination of configuration issues boils down to just the two Interfaces: **Inside** and **Outside**

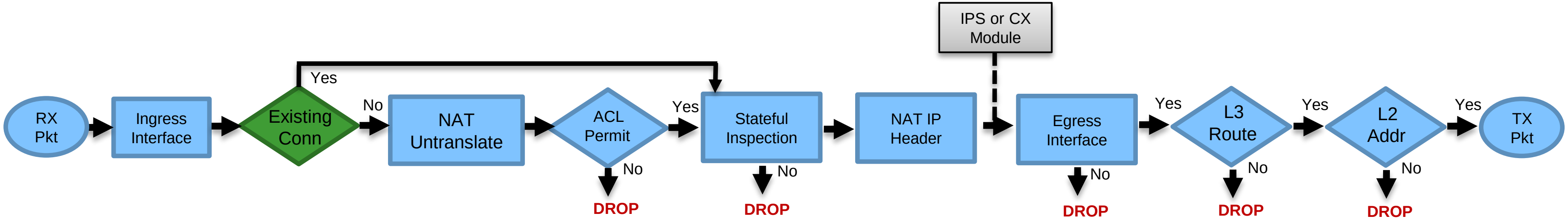
Packet Processing: Ingress Interface



- Packet arrives on ingress interface
- Input counters incremented by NIC and periodically retrieved by CPU
- Software input queue (RX ring) is an indicator of packet load
- **Overrun** counter indicates packet drops (usually packet bursts)

```
asa# show interface outside
Interface GigabitEthernet0/3 "outside", is up, line protocol is up
  Hardware is i82546GB rev03, BW 1000 Mbps, DLY 10 usec
        Auto-Duplex(Full-duplex), Auto-Speed(1000 Mbps)
        Input flow control is unsupported, output flow control is off
        MAC address 0026.0b31.36d5, MTU 1500
        IP address 148.167.254.24, subnet mask 255.255.255.128
        54365986 packets input, 19026041545 bytes, 0 no buffer
        Received 158602 broadcasts, 0 runts, 0 giants
        0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
[…]
        input queue (blocks free curr/low): hardware (255/230)
        output queue (blocks free curr/low): hardware (254/65)
```

Packet Processing: Locate Connection



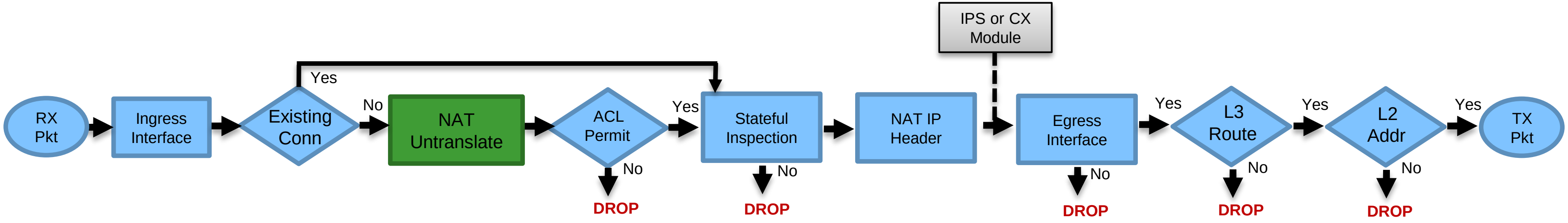
- Check first for existing connection in conn table
- If conn entry exists, bypass ACL check and process in Fastpath

```
asa# show conn
TCP out 198.133.219.25:80 in 10.1.1.9:11030 idle 0:00:04 Bytes 1293 flags UIO
```

- If no existing connection
  - TCP SYN or UDP packet, pass to ACL and other policy checks in Session Manager
  - TCP non-SYN packet, drop and log

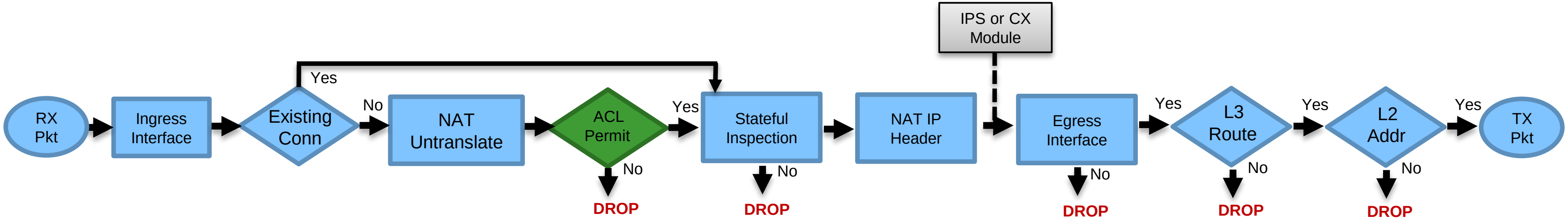
```
ASA-6-106015: Deny TCP (no connection) from 10.1.1.9/11031 to 198.133.219.25/80 flags PSH ACK  on
interface inside
```

Packet Processing: NAT Un-Translate



- Incoming packet is checked against NAT rules
- Packet is un-translated first, before ACL check
  - In **ASA 8.2** and below, incoming packet was subjected to ACL check prior to un-translation
- NAT rules can determine the egress interface at this stage

Packet Processing: ACL Check



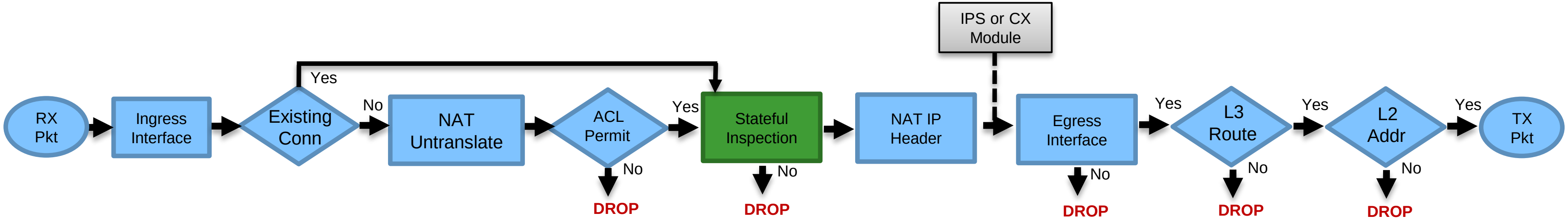
- First packet in flow is processed through ACL checks
- ACLs are **first configured** match
- First packet in flow matches ACE, incrementing hit count by one

```
asa# show access-list inside
access-list inside line 10 permit ip 10.1.1.0 255.255.255.0 any (hitcnt=1)
```

- Denied packets are dropped and logged

```
ASA-4-106023: Deny tcp src inside:10.1.1.9/11034 dst outside:198.133.219.25/80 by access-group "inside"
```

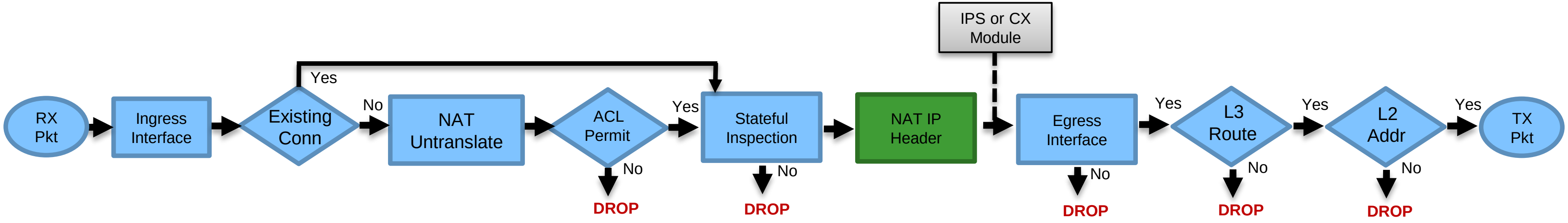
Packet Processing: Stateful Inspection



- Stateful inspection ensures protocol compliance at TCP/UDP/ICMP level
- (Optional) Customisable application inspection up to Layer 7 (FTP, SIP, and so on)
  - Rewrite embedded IP addresses, open up ACL pinholes for secondary connections
  - Additional security checks are applied to the application payload

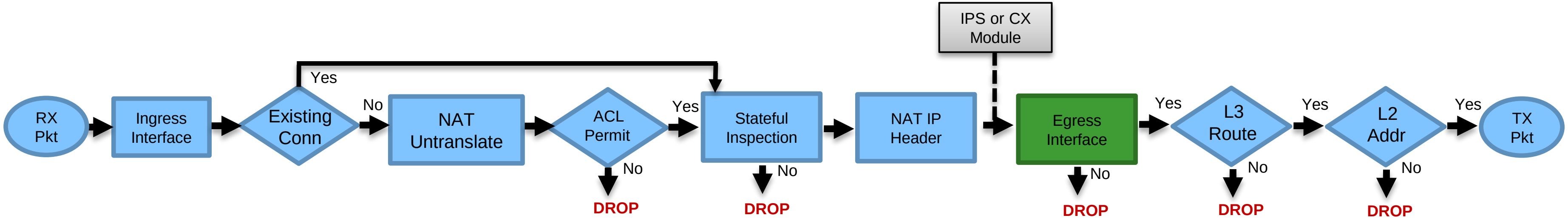
```
ASA-4-406002: FTP port command different address: 10.2.252.21(192.168.1.21) to
209.165.202.130 on interface inside

ASA-4-405104: H225 message received from outside_address/outside_port to
inside_address/inside_port before SETUP
```

Packet Processing: NAT IP Header

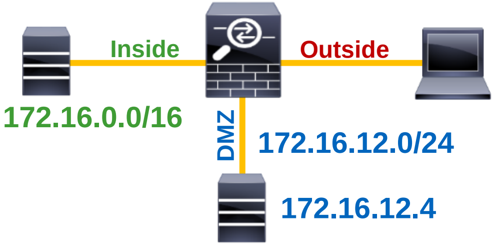


- Translate the source and destination IP addresses in the IP header
- Translate the port if performing PAT
- Update header checksums
- (Optional) Following the above, pass packet to IPS or CX module
  - Real (pre-NAT) IP address information is supplied as meta data

Packet Processing: Egress Interface



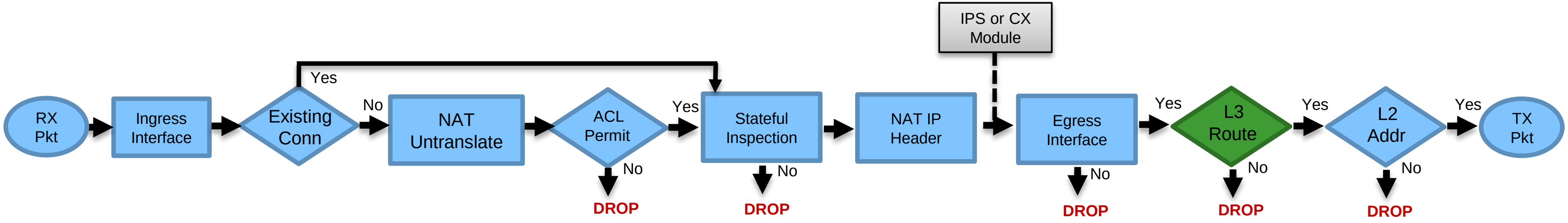
- Packet is virtually forwarded to egress interface (not forwarded to the Ethernet NIC yet)
- Egress interface is determined first by translation rules or existing conn entry, only THEN the routing table
- If NAT does not divert to the egress interface, the global routing table is consulted to determine egress interface



Packets received on **outside** and destined to **192.168.12.4** get routed to **172.16.12.4** on **inside** based on NAT configuration.

```
nat (inside,outside) source static 172.16.0.0-net 192.168.0.0-net
nat (dmz,outside) source static 172.16.12.0-net 192.168.12.0-net
```

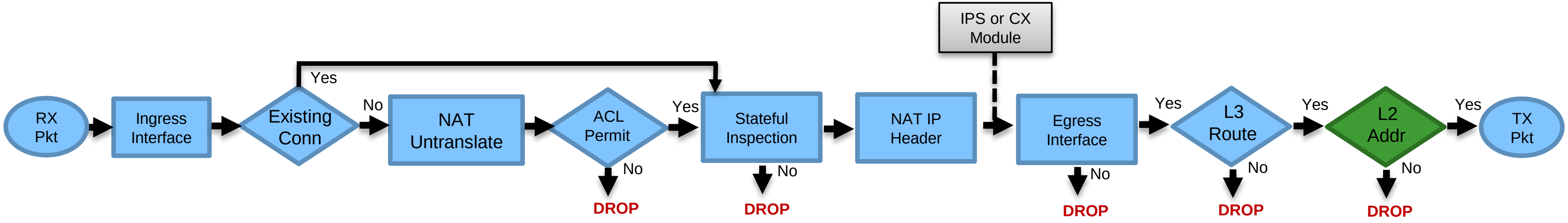
Packet Processing: L3 Route Lookup



- Once at egress interface, an interface route lookup is performed
- Only routes pointing out the egress interface are eligible
- Remember: NAT rule can forward the packet to the egress interface, even though the routing table may point to a different interface
  - If the destination is not routable out of the identified egress interface, the packet is dropped

```
%ASA-6-110003: Routing failed to locate next hop for TCP from inside:192.168.103.220/59138
to dmz:172.15.124.76/23
```

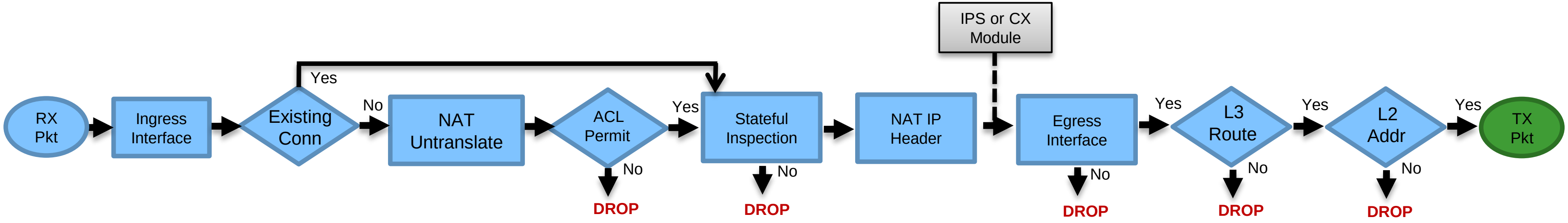
Packet Processing: L2 Address Lookup



- Once a Layer 3 route has been found, and next hop IP address identified, Layer 2 resolution is performed
  - Layer 2 rewrite of MAC header
- If Layer 2 resolution fails — **no** syslog
  - **show arp** will not display an entry for the L3 next hop
  - **debug arp** will indicate if we are not receiving an ARP reply

```
arp-req: generating request for 10.1.2.33 at interface outside
arp-req: request for 10.1.2.33 still  pending
```

Packet Processing: Transmit Packet



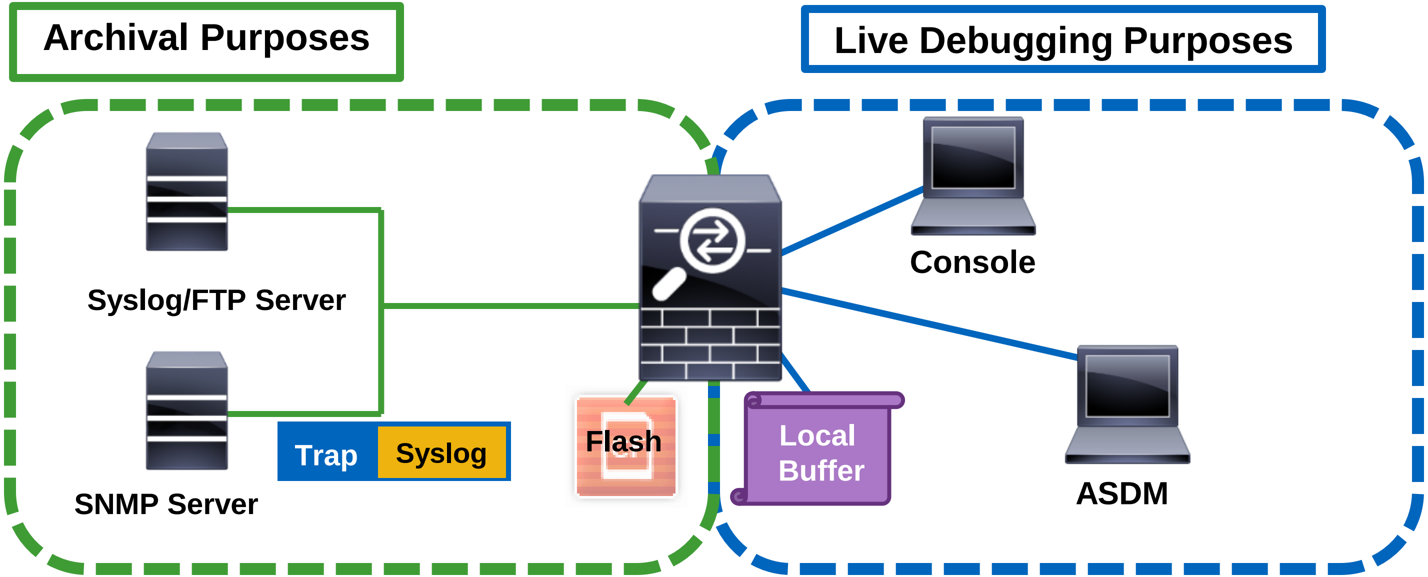
- Packet is transmitted on wire
- Interface counters will increment on interface
- **Underrun** counter indicates drops due to egress interface oversubscription
  - TX ring is full

```
asa# show interface outside
Interface GigabitEthernet0/1 "outside", is up, line protocol is up
  Hardware is i82574L rev00, BW 1000 Mbps, DLY 10 usec
           MAC address 503d.e59d.90ab, MTU 1500
           IP address 172.18.124.149, subnet mask 255.255.255.0
           …
           273399 packets output, 115316725 bytes, 80 underruns
           …
           input queue (blocks free curr/low): hardware (485/441)
           output queue (blocks free curr/low): hardware (463/0)
```

Diagnostic Messages and Outputs

Uses of Syslogs

- Primary mechanism for recording connections **to** and **through** the firewall
- The best troubleshooting tool available



Custom Syslog Levels

- Assign any syslog message to any available level
- Problem:

You want to record what exec commands are being executed on the firewall; syslog ID 111009 records this information, but by default it is at level 7 (debug)

```
ASA-7-111009: User 'johndoe' executed cmd: show run
```

The problem is we don't want to log all 1775 other syslogs that are generated at debug level

```
asa(config)# logging message 111009 level 3
```

```
ASA-3-111009: User 'johndoe' executed cmd: show run
```

Levels

0—Emergency

1—Alert

2—Critical

3—Errors

4—Warnings

5—Notifications

6—Informational

7—Debugging

NetFlow Secure Event Logging (NSEL)

- NetFlow v9 support added in **ASA 8.1+**
  - Provides a method to deliver binary logs at high speeds
  - Reduce processing overhead in printing logs
  - Combine multiple events into one NetFlow record
- FlowSets Supported:
  - Flow Creation
  - Flow Teardown
  - Flow Denied
  - Flow Update in **ASA 8.4(5)+** and **9.1(2)+**
- Remove redundant syslog messages

```
asa(config)# logging flow-export-syslogs disable
```

Case Study: Excessive Logging

```
logging enable
logging buffered debugging
logging console debugging
logging trap debugging
logging history debugging
logging host inside 192.168.1.10
logging host inside 192.168.1.11
logging host DMZ 192.168.2.121

snmp-server host inside 192.168.1.10
snmp-server host inside 192.168.1.11
snmp-server host DMZ 192.168.2.121

flow-export destination inside 192.168.1.10
flow-export destination inside 192.168.1.11
flow-export destination DMZ 192.168.2.121
```

4 logging destinations (buffer, console, SNMP, and syslog)

+

3 syslog servers

+

3 SNMP servers

+

3 Netflow collectors

×

4 messages per PAT connection (over 550 bytes)

```
%ASA-6-305011: Built dynamic TCP translation from inside:192.168.1.101/4675
outside:172.16.171.125/34605
%ASA-6-302013: Built outbound TCP connection 3367663 for outside:198.133.21
(198.133.219.25/80) to inside:192.168.1.101/4675 (172.16.171.125/34605)
%ASA-6-302014: Teardown TCP connection 3367663 for outside:198.133.219.25/8
inside:192.168.1.101/4675 duration 0:00:00 bytes 1027 TCP FINs
%ASA-6-305012: Teardown dynamic TCP translation from inside:192.168.1.101/4
outside:172.16.171.125/34605 duration 0:00:30
```

1 connection:

32 syslog messages

26+ packets sent

100K connections/sec:

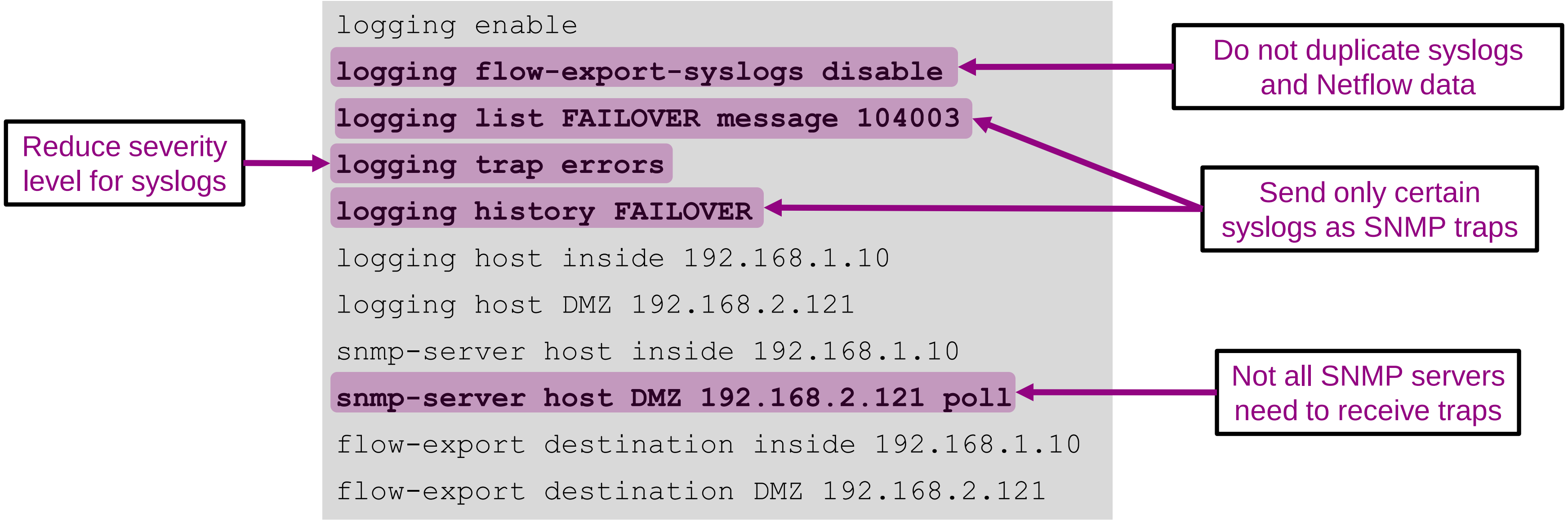
2.8Gbps

Case Study: Logging Optimisation

Not logging to buffer unless troubleshooting

Console logging is a bottleneck (low rate)

Using minimum number of syslog servers and Netflow collectors



```
logging enable
logging flow-export-syslogs disable
logging list FAILOVER message 104003
logging trap errors
logging history FAILOVER
logging host inside 192.168.1.10
logging host DMZ 192.168.2.121
snmp-server host inside 192.168.1.10
snmp-server host DMZ 192.168.2.121 poll
flow-export destination inside 192.168.1.10
flow-export destination DMZ 192.168.2.121
```

Do not duplicate syslogs and Netflow data

Reduce severity level for syslogs

Send only certain syslogs as SNMP traps

Not all SNMP servers need to receive traps

Debug Commands

- Debugs should not be the first choice to troubleshoot a problem
- Debugs can **negatively** impact the CPU complex and affect performance
- Most debugs are not conditional
- Know how much traffic of the matching type is passing through the firewall before enabling the respective debug

Show Output Filters

See Appendix

- Filters limit the output of **show** commands to only what you want to see
- Use the pipe character “**|**” at the end of **show <command>** followed by
  - **begin** Start displaying the output beginning at the first match of the RegEx, and continue to display the remaining output
  - **include** Display any line that matches the RegEx
  - **exclude** Display any line that does not match the RegEx
  - **grep** Same as include
  - **grep -v** Same as exclude
  - **redirect** Send output to a file (flash, tftp, ftp…)
  - **append** Append output to an existing file (flash, tftp, ftp…)

```
show <cmd> | begin|include|exclude|grep|redirect|append [-v] <regular_exp>
```

Monitoring CPU Usage

- ASA starts dropping packets when aggregated CPU usage reaches 100%

```
asa# show cpu detail
Break down of per-core data path versus control point cpu usage:
Core          5 sec              1 min              5 min
Core 0        0.0 (0.0 + 0.0)    0.0 (0.0 + 0.0)    0.0 (0.0 + 0.0)
Core 1        0.0 (0.0 + 0.0)    0.0 (0.0 + 0.0)    0.0 (0.0 + 0.0)
Core 2        0.0 (0.0 + 0.0)    0.0 (0.0 + 0.0)    0.0 (0.0 + 0.0)
Core 3        0.0 (0.0 + 0.0)    0.0 (0.0 + 0.0)    0.0 (0.0 + 0.0)
Core 4        0.0 (0.0 + 0.0)    0.0 (0.0 + 0.0)    0.0 (0.0 + 0.0)
Core 5        0.0 (0.0 + 0.0)    0.0 (0.0 + 0.0)    0.0 (0.0 + 0.0)
Core 6        0.0 (0.0 + 0.0)    0.0 (0.0 + 0.0)    0.0 (0.0 + 0.0)
Core 7        0.0 (0.0 + 0.0)    0.0 (0.0 + 0.0)    0.0 (0.0 + 0.0)

Current control point elapsed versus the maximum control point elapsed for:
      5 seconds = 83.3%; 1 minute: 83.3%; 5 minutes: 83.3%

CPU utilization of external processes for:
      5 seconds = 0.0%; 1 minute: 0.0%; 5 minutes: 0.0%

Total CPU utilization for:
      5 seconds = 0.1%; 1 minute: 0.0%; 5 minutes: 0.0%
```

Each CPU core processes packets independently, so each can load up to 100%

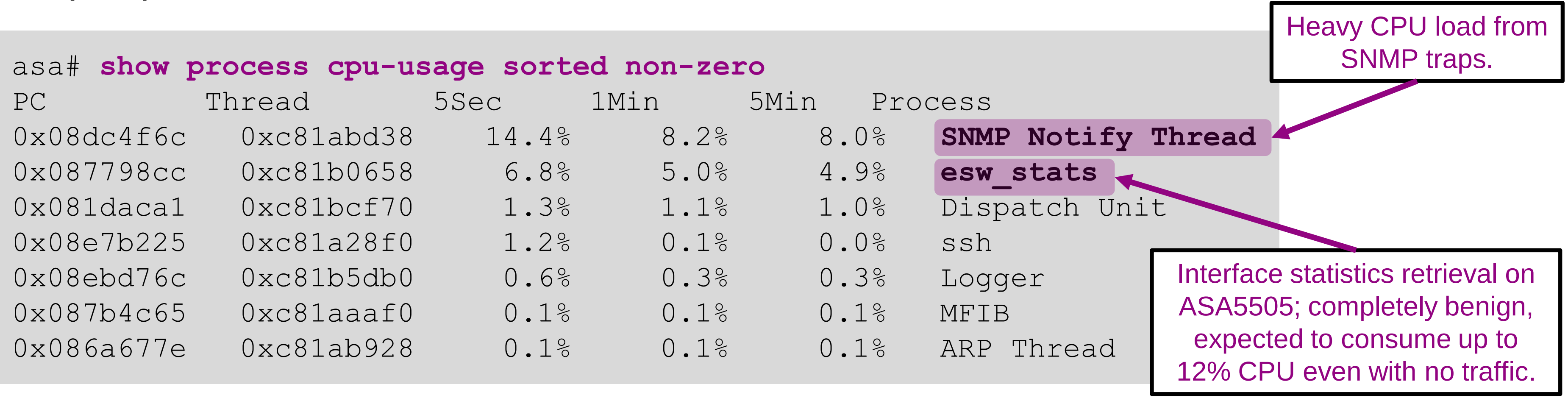
% load reported for each interval: **Total (Data Path + Control Path)**

Control Path load over each interval is compared to the high watermark over uptime. **100%** means steady load, not oversubscription.

Aggregated utilisation across all cores, same as in **show cpu**

CPU Utilisation by Processes

- **show processes cpu-usage** command displays the amount of CPU used on a per-process basis for the last 5 sec, 1 min, and 5 min



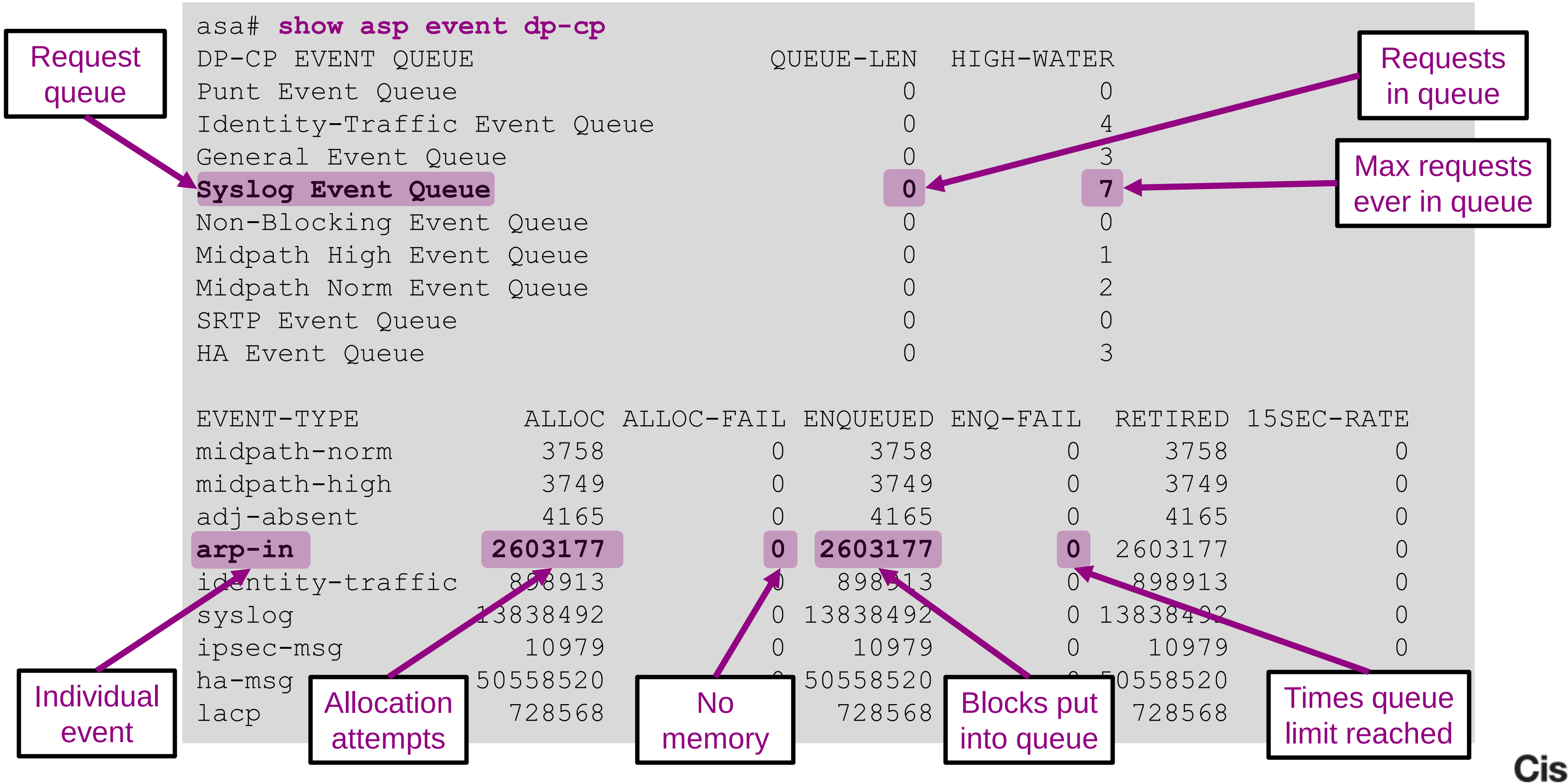
```
asa# show process cpu-usage sorted non-zero
PC           Thread        5Sec     1Min     5Min    Process
0x08dc4f6c   0xc81abd38    14.4%     8.2%     8.0%   SNMP Notify Thread
0x087798cc   0xc81b0658     6.8%     5.0%     4.9%   esw_stats
0x081daca1   0xc81bcf70     1.3%     1.1%     1.0%   Dispatch Unit
0x08e7b225   0xc81a28f0     1.2%     0.1%     0.0%   ssh
0x08ebd76c   0xc81b5db0     0.6%     0.3%     0.3%   Logger
0x087b4c65   0xc81aaaf0     0.1%     0.1%     0.1%   MFIB
0x086a677e   0xc81ab928     0.1%     0.1%     0.1%   ARP Thread
```

Heavy CPU load from SNMP traps.

Interface statistics retrieval on ASA5505; completely benign, expected to consume up to 12% CPU even with no traffic.

- Use **cpu profile** under TAC supervision for per-function load granularity

Multi-Core ASA Control Path Queue



```
asa# show asp event dp-cp
DP-CP EVENT QUEUE                  QUEUE-LEN  HIGH-WATER
Punt Event Queue                           0           0
Identity-Traffic Event Queue               0           4
General Event Queue                        0           3
Syslog Event Queue                         0           7
Non-Blocking Event Queue                   0           0
Midpath High Event Queue                   0           1
Midpath Norm Event Queue                   0           2
SRTP Event Queue                           0           0
HA Event Queue                             0           3

EVENT-TYPE          ALLOC ALLOC-FAIL ENQUEUED ENQ-FAIL  RETIRED 15SEC-RATE
midpath-norm         3758          0     3758        0     3758          0
midpath-high         3749          0     3749        0     3749          0
adj-absent           4165          0     4165        0     4165          0
arp-in            2603177          0  2603177        0  2603177          0
identity-traffic   898913          0   898913        0   898913          0
syslog           13838492          0 13838492        0 13838492          0
ipsec-msg           10979          0    10979        0    10979          0
ha-msg           50558520          0 50558520        0 50558520
lacp               728568               728568             728568
```

Traffic Rates

```
asa# show traffic
[…]
TenGigabitEthernet5/1:
        received (in 2502.440 secs):
                99047659 packets        130449274327 bytes
                39580 pkts/sec  52128831 bytes/sec
        transmitted (in 2502.440 secs):
                51704620 packets        3581723093 bytes
                20661 pkts/sec  1431292 bytes/sec
      1 minute input rate 144028 pkts/sec,  25190735 bytes/sec
      1 minute output rate 74753 pkts/sec,  5145896 bytes/sec
      1 minute drop rate, 0 pkts/sec
      5 minute input rate 131339 pkts/sec, 115953675 bytes/sec
      5 minute output rate 68276 pkts/sec,  4748861 bytes/sec
      5 minute drop rate, 0 pkts/sec
```

Uptime statistics is useful to determine historical average packet size and rates:
52128831 B/sec / **39580 pkts/sec** = ~**1317 B/packet**

One-minute average is useful to detect bursts and small packets:
25190735 B/sec / **144028 pkts/sec** = ~**174 B/packet**

Xlate Table

- **show xlate** displays information about NAT translations through the ASA
  - Second biggest memory consumer after conn table, no hardcoded size limit
- You can limit the output to just the **local** or **global** IP

```
asa# show xlate local 10.2.1.2
5014 in use, 5772 most used
TCP PAT from inside:192.168.103.220/57762 to outside:10.2.1.2/43756 flags ri
idle 0:00:00 timeout 0:00:30
TCP PAT from inside:192.168.103.220/57761 to outside:10.2.1.2/54464 flags ri
idle 0:00:00 timeout 0:00:30
```

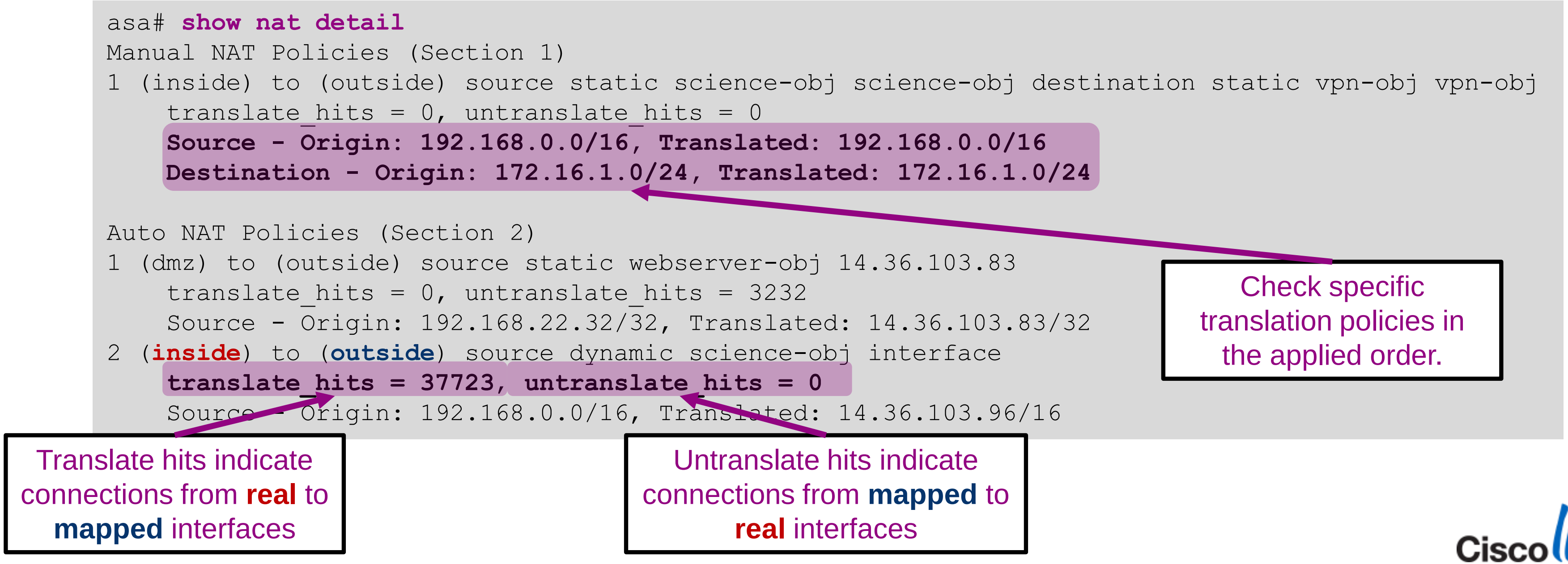
- Depleted NAT/PAT pools may cause connectivity issues

```
asa# show nat pool
TCP PAT pool outside, address 10.2.1.2, range 1-511, allocated 1
TCP PAT pool outside, address 10.2.1.2, range 512-1023, allocated 0
TCP PAT pool outside, address 10.2.1.2, range 1024-65535, allocated 64102
```

Detailed NAT Information

TAC Tip

- **show nat** displays information about the NAT table of the ASA
  - **detail** keyword will display object definitions
  - Watch the hit counts for policies that are not matching traffic



```
asa# show nat detail
Manual NAT Policies (Section 1)
1 (inside) to (outside) source static science-obj science-obj destination static vpn-obj vpn-obj
    translate_hits = 0, untranslate_hits = 0
    Source - Origin: 192.168.0.0/16, Translated: 192.168.0.0/16
    Destination - Origin: 172.16.1.0/24, Translated: 172.16.1.0/24

Auto NAT Policies (Section 2)
1 (dmz) to (outside) source static webserver-obj 14.36.103.83
    translate_hits = 0, untranslate_hits = 3232
    Source - Origin: 192.168.22.32/32, Translated: 14.36.103.83/32
2 (inside) to (outside) source dynamic science-obj interface
    translate_hits = 37723, untranslate_hits = 0
    Source - Origin: 192.168.0.0/16, Translated: 14.36.103.96/16
```

Check specific translation policies in the applied order.

Translate hits indicate connections from **real** to **mapped** interfaces

Untranslate hits indicate connections from **mapped** to **real** interfaces

Connection Table

```
asa# show conn detail
2 in use, 64511 most used
Flags: A - awaiting inside ACK to SYN, a - awaiting outside ACK to SYN,
       B - initial SYN from outside, b - TCP state-bypass or nailed,
       C - CTIQBE media, c - cluster centralized,
       D - DNS, d - dump, E - outside back connection, F - outside FIN, f - inside FIN,
       G - group, g - MGCP, H - H.323, h - H.225.0, I - inbound data,
       i - incomplete, J - GTP, j - GTP data, K - GTP t3-response
       k - Skinny media, M - SMTP data, m - SIP media, n - GUP
       O - outbound data, P - inside back connection, p - Phone-proxy TFTP connection,
       q - SQL*Net data, R - outside acknowledged FIN,
       R - UDP SUNRPC, r - inside acknowledged FIN, S - awaiting inside SYN,
       s - awaiting outside SYN, T - SIP, t - SIP transient, U - up,
       V - VPN orphan, W - WAAS,
       X - inspected by service module,
       x - per session, Y - director stub flow, y - backup stub flow,
       Z - Scansafe redirection, z - forwarding stub flow

TCP outside:198.133.219.25/80 dmz:10.9.9.3/4101,
    flags UIO, idle 8s, uptime 10s, timeout 1h, bytes 127
UDP outside:172.18.124.1/123 dmz:10.1.1.9/123,
    flags -, idle 15s, uptime 16s, timeout 2m, bytes 1431
```

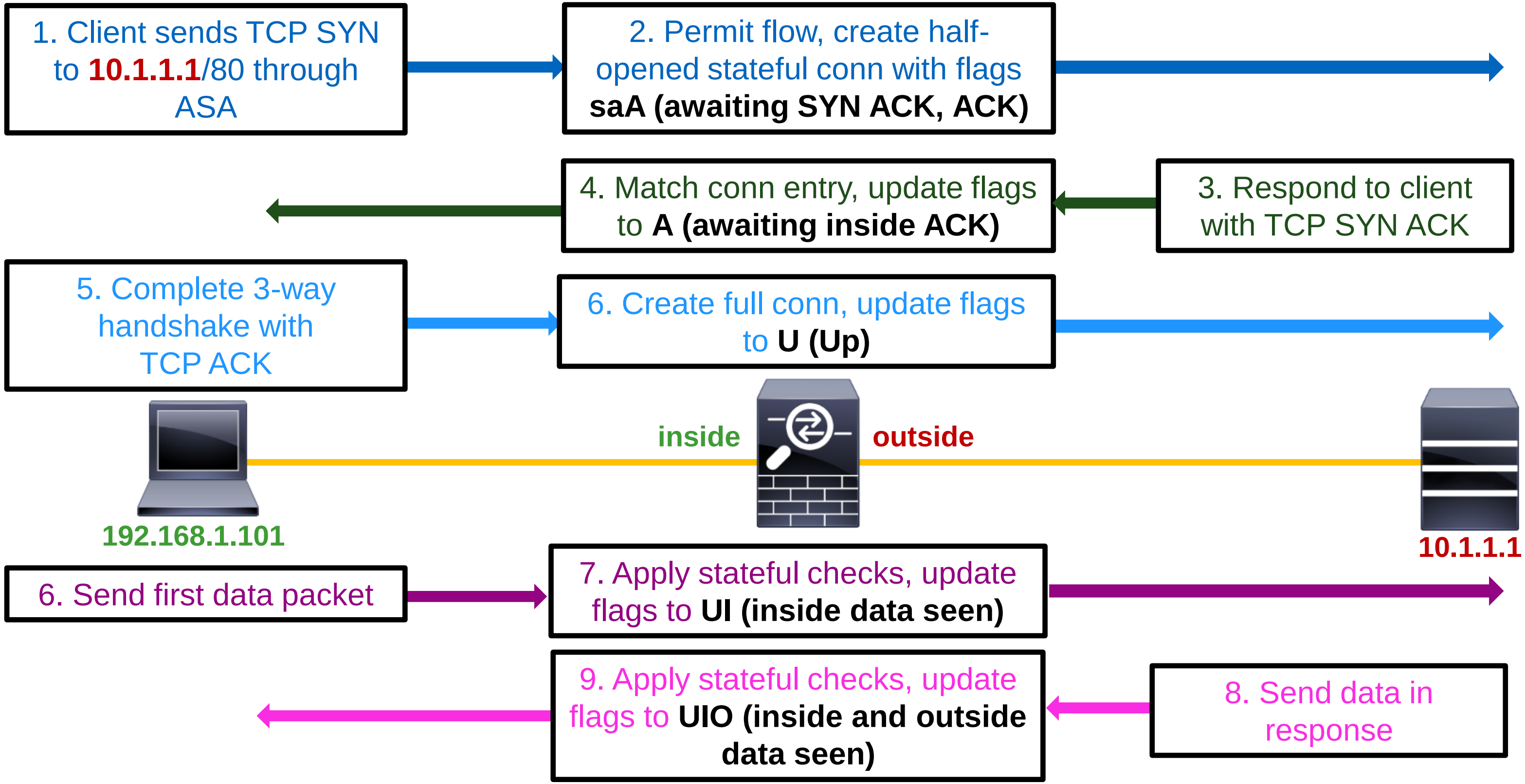
Narrow down the output with **show conn address <ip>**

Bidirectional byte count; use NSEL to report each direction separately.

Conn flags indicate current state

detail option adds uptime and timeout information

Example: Connection Establishment



1. Client sends TCP SYN to **10.1.1.1**/80 through ASA

2. Permit flow, create half-opened stateful conn with flags **saA (awaiting SYN ACK, ACK)**

3. Respond to client with TCP SYN ACK

4. Match conn entry, update flags to **A (awaiting inside ACK)**

5. Complete 3-way handshake with TCP ACK

6. Create full conn, update flags to **U (Up)**

6. Send first data packet

7. Apply stateful checks, update flags to **UI (inside data seen)**

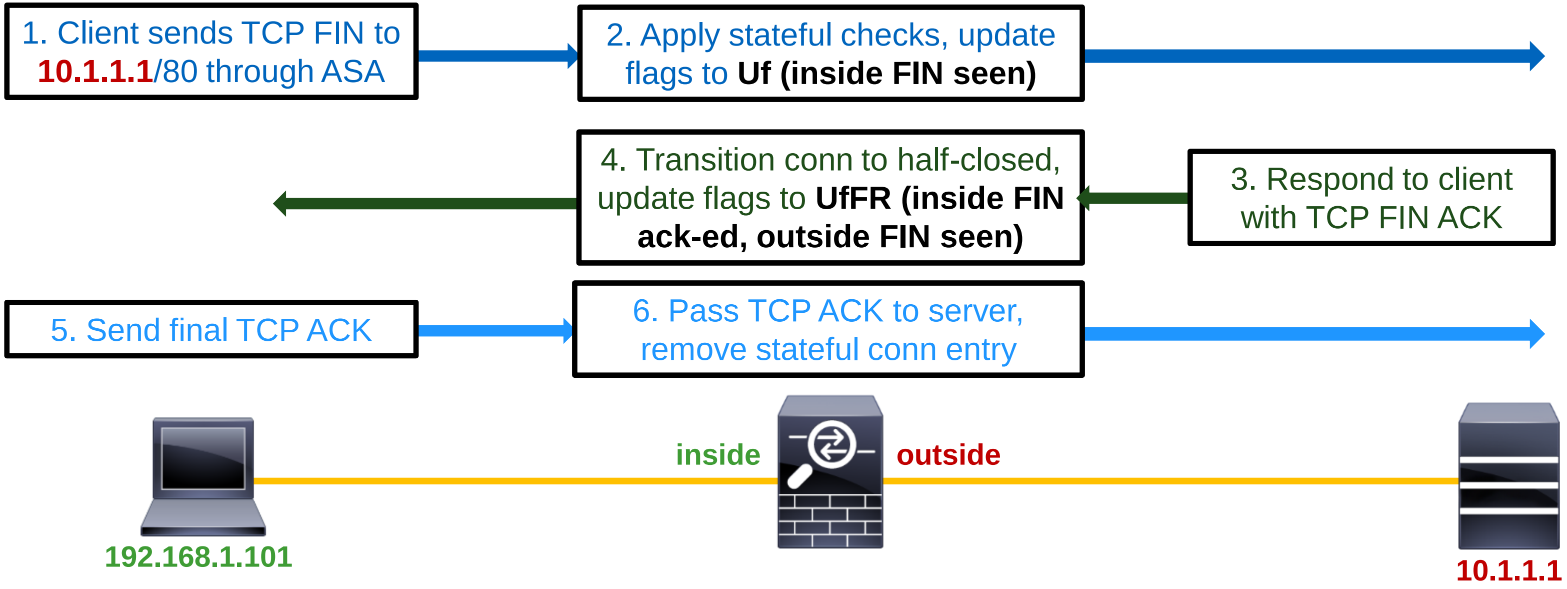
8. Send data in response

9. Apply stateful checks, update flags to **UIO (inside and outside data seen)**

```
TCP outside 10.1.1.1:80 inside 192.168.1.101:50141, idle 0:00:00, bytes 153, flags UIO
```

Example: Connection Termination

```
TCP outside 10.1.1.1:80 inside 192.168.1.101:50141, idle 0:00:00, bytes 153, flags UIO
```



1. Client sends TCP FIN to **10.1.1.1**/80 through ASA

2. Apply stateful checks, update flags to **Uf (inside FIN seen)**

3. Respond to client with TCP FIN ACK

4. Transition conn to half-closed, update flags to **UfFR (inside FIN ack-ed, outside FIN seen)**

5. Send final TCP ACK

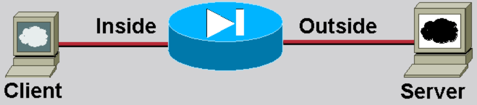
6. Pass TCP ACK to server, remove stateful conn entry

Connection Flags

For your reference

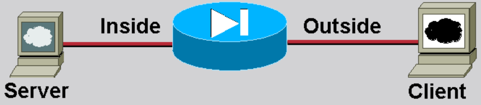
Outbound Connection

TCP Flags		FW Flags
SYN	→	saA
SYN+ACK	←	A
ACK	→	U
Inbound Data	←	UI
Outbound Data	→	UIO
FIN	→	Uf
FIN+ACK	←	UfFR
ACK	→	UfFRr



Inbound Connection

TCP Flags		FW Flags
SYN	←	SaAB
SYN+ACK	→	aB
ACK	←	UB
Inbound Data	←	UIB
Outbound Data	→	UIOB
FIN	←	UBF
FIN+ACK	→	UBfFr
ACK	←	UBfFRr



TCP Connection Termination Reasons

- If a TCP flow was built through the ASA, it will **always** log a teardown reason
- TCP teardown message is logged at level 6 (informational) by default
- If you are having problems abnormal connection termination, temporally increase your logging level (or change the syslog level, and check the teardown reason

What do these termination reasons mean in the Teardown TCP connection syslog?

```
%ASA-6-302014: Teardown TCP connection 90 for outside:10.1.1.1/80 to inside:192.168.1.101/1107 duration 0:00:30 bytes 0
SYN Timeout
```

```
%ASA-6-302014: Teardown TCP connection 3681 for DMZ:172.16.171.125/21 to inside:192.168.1.110/24245 duration 0:01:03
bytes 12504 TCP Reset-O
```

TCP Connection Termination Reasons

For your reference

Reason	Description
Conn-Timeout	Connection Ended Because It Was Idle Longer Than the Configured Idle Timeout
Deny Terminate	Flow Was Terminated by Application Inspection
Failover Primary Closed	The Standby Unit in a Failover Pair Deleted a Connection Because of a Message Received from the Active Unit
FIN Timeout	Force Termination After Ten Minutes Awaiting the Last ACK or After Half-Closed Timeout
Flow Closed by Inspection	Flow Was Terminated by Inspection Feature
Flow Terminated by IPS	Flow Was Terminated by IPS
Flow Reset by IPS	Flow Was Reset by IPS
Flow Terminated by TCP Intercept	Flow Was Terminated by TCP Intercept
Invalid SYN	SYN Packet Not Valid
Idle Timeout	Connection Timed Out Because It Was Idle Longer than the Timeout Value
IPS Fail-Close	Flow Was Terminated Due to IPS Card Down
SYN Control	Back Channel Initiation from Wrong Side

TCP Connection Termination Reasons

For your reference

Reason	Description
SYN Timeout	Force Termination After Twenty Seconds Awaiting Three-Way Handshake Completion
TCP Bad Retransmission	Connection Terminated Because of Bad TCP Retransmission
TCP Fins	Normal Close Down Sequence
TCP Invalid SYN	Invalid TCP SYN Packet
TCP Reset-I	TCP Reset Was Sent From the Inside Host
TCP Reset-O	TCP Reset Was Sent From the Outside Host
TCP Segment Partial Overlap	Detected a Partially Overlapping Segment
TCP Unexpected Window Size Variation	Connection Terminated Due to a Variation in the TCP Window Size
Tunnel Has Been Torn Down	Flow Terminated Because Tunnel Is Down
Unauth Deny	Connection Denied by URL Filtering Server
Unknown	Catch-All Error
Xlate Clear	User Executed the ‘Clear Xlate’ Command

Local Host Table

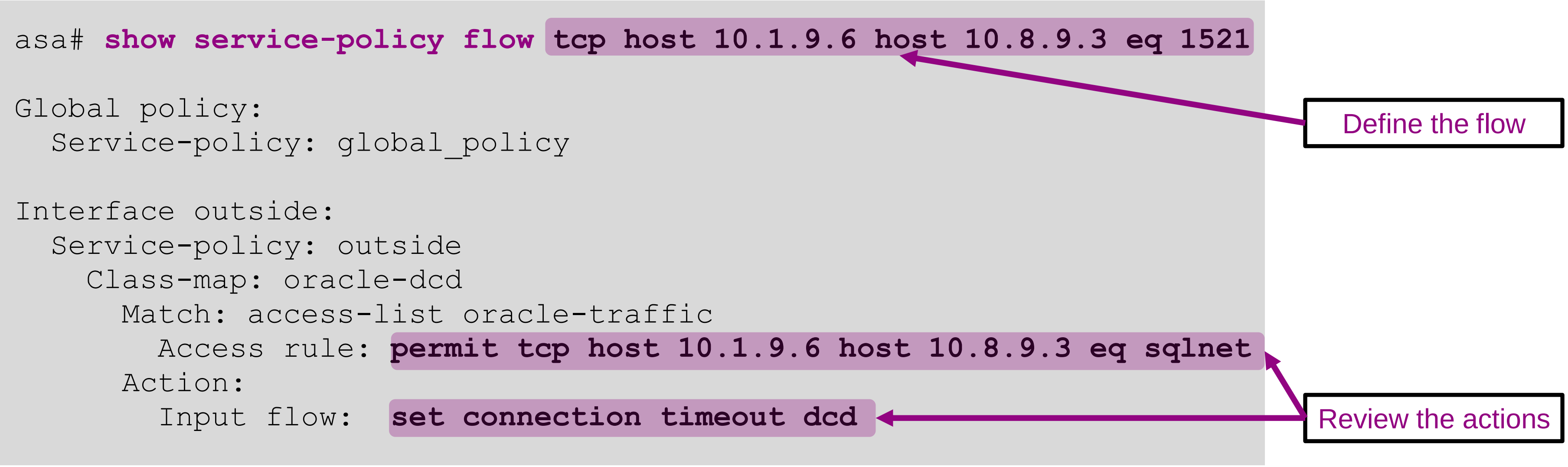
- A local-host entry is created for every IP tracked by the ASA
- It groups xlates, connections, and AAA information
- Useful for monitoring connections terminating on servers or offending clients

```
asa# show local-host detail connection tcp 50
Interface dmz: 0 active, 0 maximum active, 0 denied
Interface inside: 1 active, 1 maximum active, 0 denied
local host: <192.168.103.220>,
    TCP flow count/limit = 798/unlimited
    TCP embryonic count to host = 0
    TCP intercept watermark = unlimited
    UDP flow count/limit = 0/unlimited
 Conn:
    TCP outside:172.18.124.76/80 inside:192.168.103.220/34078,
        flags UO, idle 0s, uptime 0s, timeout 30s, bytes 0
    TCP outside:172.18.124.76/80 inside:192.168.103.220/34077,
        flags UO, idle 0s, uptime 0s, timeout 30s, bytes 0
   (output truncated)
```

Only display hosts that have more than 50 active TCP connections.

Service Policy Information

- **show service-policy** command displays high level Modular Policy Framework (MPF) counters
- Use **show service-policy flow** to see what MPF policies will match a flow



```
asa# show service-policy flow tcp host 10.1.9.6 host 10.8.9.3 eq 1521

Global policy:
  Service-policy: global_policy

Interface outside:
  Service-policy: outside
    Class-map: oracle-dcd
      Match: access-list oracle-traffic
        Access rule: permit tcp host 10.1.9.6 host 10.8.9.3 eq sqlnet
      Action:
        Input flow:  set connection timeout dcd
```

Define the flow

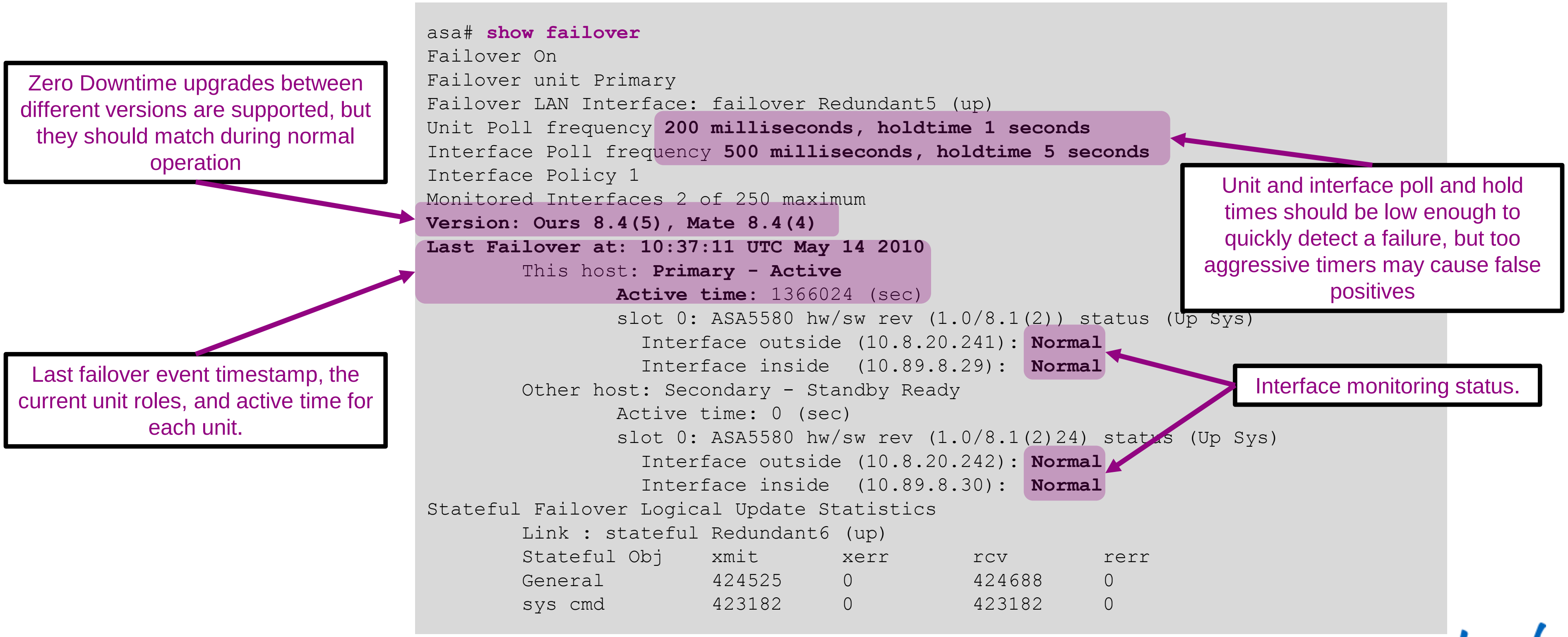
Review the actions

Accelerated Security Path (ASP)

- Packets and flows dropped in the ASP will increment a counter
  - Frame drop counters are per packet
  - Flow drops are per flow
- See command reference under **show asp drop** for full list of counters

```
asa# show asp drop

Frame drop:
   Invalid encapsulation (invalid-encap)                          10897
   Invalid tcp length (invalid-tcp-hdr-length)                     9382
   Invalid udp length (invalid-udp-length)                           10
   No valid adjacency (no-adjacency)                               5594
   No route to host (no-route)                                     1009
   Reverse-path verify failed (rpf-violated)                         15
   Flow is denied by access rule (acl-drop)                    25247101
   First TCP packet not SYN (tcp-not-syn)                         36888
   Bad TCP flags (bad-tcp-flags)                                  67148
   TCP option list invalid (tcp-bad-option-list)                    731
   TCP MSS was too large (tcp-mss-exceeded)                       10942
   Bad TCP Checksum (bad-tcp-cksum)                                 893

…
```

Verifying Failover Operation



```
asa# show failover
Failover On
Failover unit Primary
Failover LAN Interface: failover Redundant5 (up)
Unit Poll frequency 200 milliseconds, holdtime 1 seconds
Interface Poll frequency 500 milliseconds, holdtime 5 seconds
Interface Policy 1
Monitored Interfaces 2 of 250 maximum
Version: Ours 8.4(5), Mate 8.4(4)
Last Failover at: 10:37:11 UTC May 14 2010
        This host: Primary - Active
                Active time: 1366024 (sec)
                slot 0: ASA5580 hw/sw rev (1.0/8.1(2)) status (Up Sys)
                  Interface outside (10.8.20.241): Normal
                  Interface inside  (10.89.8.29):  Normal
        Other host: Secondary - Standby Ready
                Active time: 0 (sec)
                slot 0: ASA5580 hw/sw rev (1.0/8.1(2)24) status (Up Sys)
                  Interface outside (10.8.20.242): Normal
                  Interface inside  (10.89.8.30):  Normal
Stateful Failover Logical Update Statistics
        Link : stateful Redundant6 (up)
        Stateful Obj    xmit       xerr       rcv        rerr
        General         424525     0          424688     0
        sys cmd         423182     0          423182     0
```

Zero Downtime upgrades between different versions are supported, but they should match during normal operation

Last failover event timestamp, the current unit roles, and active time for each unit.

Unit and interface poll and hold times should be low enough to quickly detect a failure, but too aggressive timers may cause false positives

Interface monitoring status.

What to Do After a Failover Event

- Always check the syslogs to determine root cause
  - Example: switch port failed on inside interface of active firewall

Syslogs from Primary (Active) ASA

```
ASA-4-411002: Line protocol on Interface inside, changed state to down
ASA-1-105007: (Primary) Link status ‘Down’ on interface 1
ASA-1-104002: (Primary) Switching to STNDBY—interface check, mate is healthier
```

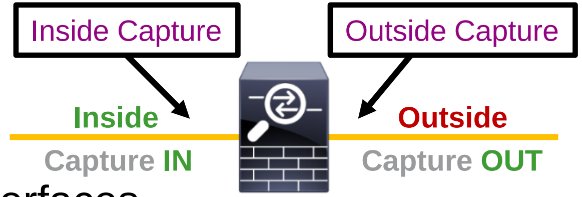
Syslogs from Secondary (Standby) ASA

```
ASA-1-104001: (Secondary) Switching to ACTIVE—mate want me Active
```

- Check **show failover history** to see the state transition times and reasons
  - Use **show cluster history** with clustering

Troubleshooting Tools

Packet Capture



- In-line capability to record packets passing through ASA
- Two key steps in troubleshooting with captures
  - Apply capture under unique name to ingress and egress interfaces
  - Define the traffic that you want to capture, use pre-NAT “on the wire” information
  - Tcpdump-like format for displaying captured packets on the box

```
asa# capture OUT interface outside match ip any host 172.18.124.1
asa# capture IN interface inside match ip any host 172.18.124.1
asa# show capture IN

4 packets captured

   1: 10:51:26.139046       802.1Q vlan#10 P0 172.18.254.46 > 172.18.124.1: icmp: echo request
   2: 10:51:26.139503       802.1Q vlan#10 P0 172.18.124.1 > 172.18.254.46: icmp: echo reply
   3: 10:51:27.140739       802.1Q vlan#10 P0 172.18.254.46 > 172.18.124.1: icmp: echo request
   4: 10:51:27.141182       802.1Q vlan#10 P0 172.18.124.1 > 172.18.254.46: icmp: echo reply
4 packets shown
asa# no capture IN
```

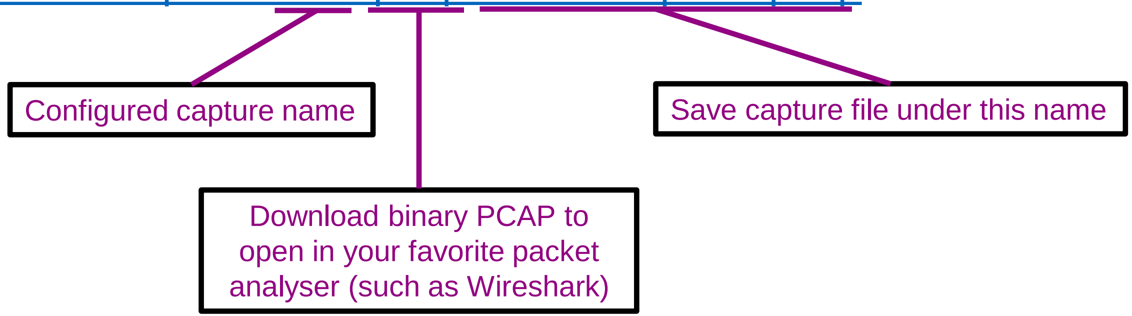
Unlike ACL, **match** covers both directions of the flow

Remember to remove the captures when done with troubleshooting

Packet Capture

- Capture buffer maintained in RAM (512KB by default)
  - Stops capturing when full by default, **circular** option available
- Default recorded packet length is 1518 bytes
- May elevate CPU utilisation on multiple-core ASA when applied
- Copy captures off via TFTP or retrieve through HTTPS with your web browser
  - Do this before removing the capture with **no capture**

    https://x.x.x.x/admin/capture/OUT/pcap/outsidecapture.pcap

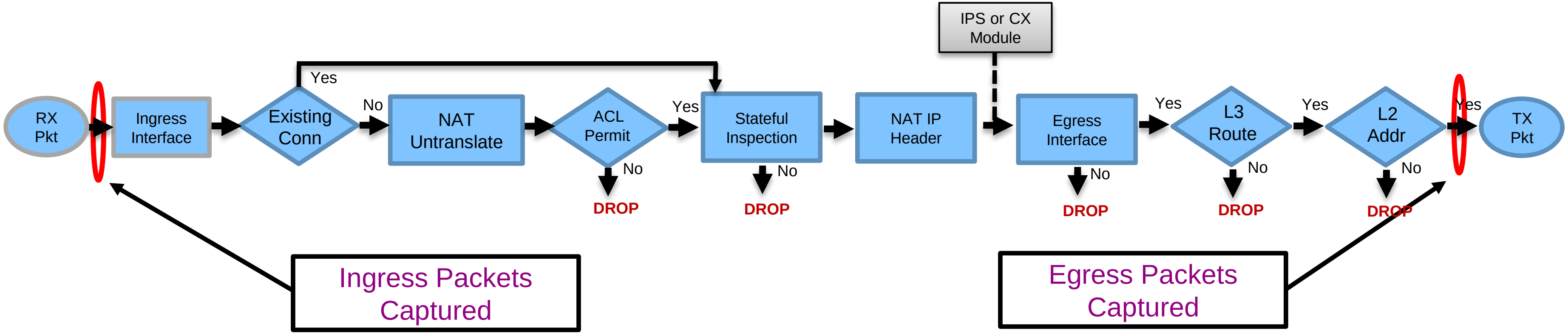


Configured capture name

Save capture file under this name

Download binary PCAP to open in your favorite packet analyser (such as Wireshark)

Where Packets Are Captured in Packet Flow



- Packets are captured at the first and last points they can be in the flow
- Ingress packets are captured **before** any packet processing
- Egress packets are captured **after** all processing
  - Transit packets show the destination MAC address rewritten
  - Self-sourced packets may show an empty MAC address (0000.0000.0000)

Capturing ASP Drops

- Capture all frames dropped in the ASP

```
asa# capture drops type asp-drop all
```

- Capture all frames with a specific drop reason

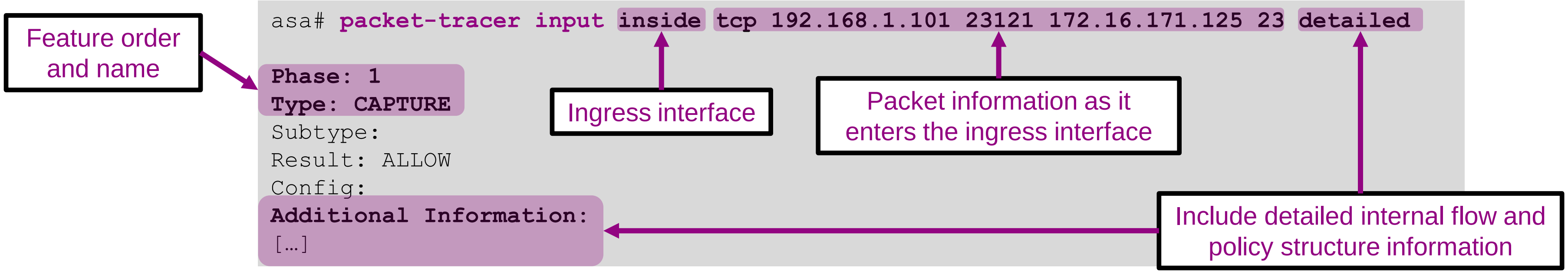
```
asa# capture drops type asp-drop tcp-not-syn
```

```
asa# capture drop type asp-drop ?

  acl-drop                        Flow is denied by configured
  rule
  all                             All packet drop reasons
  bad-crypto                      Bad crypto return in packet
  bad-ipsec-natt                  Bad IPSEC NATT packet
  bad-ipsec-prot                  IPSEC not AH or ESP
  bad-ipsec-udp                   Bad IPSEC UDP packet
  bad-tcp-cksum                   Bad TCP checksum
  bad-tcp-flags                   Bad TCP flags
```

- ASP flow drops are non-atomic and cannot be captured

Packet Tracer

- Unique capability to record the path of a specially tagged packet through ASA
  - Best way to understand the packet path in the specific software version
- Inject a simulated packet to analyse the behaviour and validate configuration



```
asa# packet-tracer input inside tcp 192.168.1.101 23121 172.16.171.125 23 detailed

Phase: 1
Type: CAPTURE
Subtype:
Result: ALLOW
Config:
Additional Information:
[…]
```

Feature order and name

Ingress interface

Packet information as it enters the ingress interface

Include detailed internal flow and policy structure information

Sample Packet Tracer Output

```
asa# packet-tracer input outside tcp 172.18.124.66 1234 172.18.254.139 3389

Phase: 1
Type: CAPTURE
Subtype:
Result: ALLOW
Config:
Additional Information:
MAC Access list

Phase: 2
Type: ACCESS-LIST
Subtype:
Result: ALLOW
Config:
Implicit Rule
Additional Information:
MAC Access list

Phase: 3
Type: UN-NAT
Subtype: static
Result: ALLOW
Config:
nat (outside,dmz) source dynamic any interface destination static interface Win7-vm service rdp-outside rdp-outside
Additional Information:
NAT divert to egress interface dmz
Untranslate 172.18.254.139/3389 to 192.168.103.221/3389
…….
```

Sample Packet Tracer Output

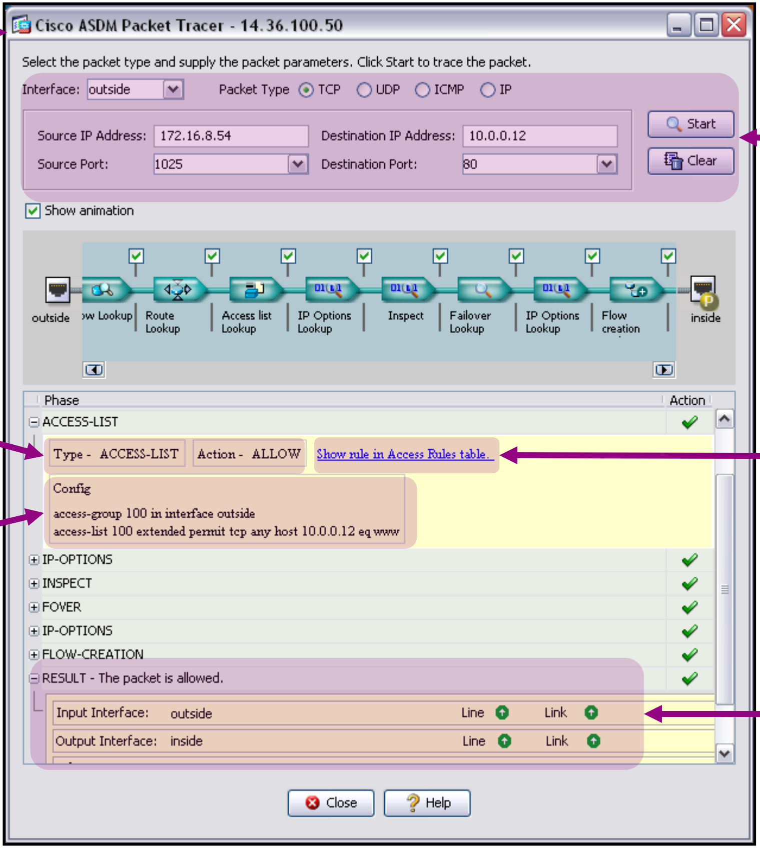
```
Phase: 4
Type: ACCESS-LIST
Subtype: log
Result: ALLOW
Config:
access-group outside_in in interface outside
access-list outside_in extended permit tcp any any eq 3389
Additional Information:
……
Phase: 8
Type: NAT
Subtype:
Result: ALLOW
Config:
nat (outside,dmz) source dynamic any interface destination static interface Win7-vm service rdp-outside rdp-outside
Additional Information:
Dynamic translate 172.18.124.66/1234 to 192.168.103.221/1234
……
Phase: 12
Type: FLOW-CREATION
Subtype:
Result: ALLOW
Config:
Additional Information:
New flow created with id 16538274, packet dispatched to next module
```

Packet Tracer in ASDM

Launch from **Tools > Packet Tracer**

Feature type and resulting action

Associated configuration



Cisco ASDM Packet Tracer - 14.36.100.50

Select the packet type and supply the packet parameters. Click Start to trace the packet.

Interface: outside

Packet Type ● TCP ○ UDP ○ ICMP ○ IP

Source IP Address: 172.16.8.54 Destination IP Address: 10.0.0.12

Source Port: 1025 Destination Port: 80

☑ Show animation

Phase	Action
⊟ ACCESS-LIST	✓
Type - ACCESS-LIST Action - ALLOW Show rule in Access Rules table.	
Config access-group 100 in interface outside access-list 100 extended permit tcp any host 10.0.0.12 eq www	
⊞ IP-OPTIONS	✓
⊞ INSPECT	✓
⊞ FOVER	✓
⊞ IP-OPTIONS	✓
⊞ FLOW-CREATION	✓
⊟ RESULT - The packet is allowed.	✓
Input Interface: outside Line ● Link ●	
Output Interface: inside Line ● Link ●	

Define simulated packet

Direct link to edit policy

Final outcome (allowed or dropped) and egress interface information

Packet Tracer: Tracing Captured Packet

- Enable packet tracer within an internal packet capture

```
asa# capture IN interface inside trace trace-count 20 match tcp any any eq
```

Trace inbound packets only

Traced packet count per capture (50 by default)

- Find the packet that you want to trace in the capture

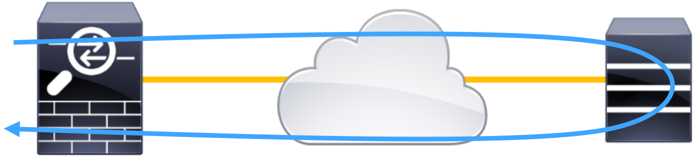
```
asa# show capture inside
  68 packets captured
  1: 15:22:47.581116 10.1.1.2.31746 > 198.133.219.25.80: S
  2: 15:22:47.583465 198.133.219.25.80 > 10.1.1.2.31746: S ack
  3: 15:22:47.585052 10.1.1.2.31746 > 198.133.219.25.80: . ack
  4: 15:22:49.223728 10.1.1.2.31746 > 198.133.219.25.80: P ack
  5: 15:22:49.223758 198.133.219.25.80 > 10.1.1.2.31746: . Ack
     ...
```

- Select that packet to show the tracer results

```
asa# show capture inside trace packet-number 4
```

TCP Ping

- Powerful troubleshooting tool added in **ASA 8.4(1)+**
- Verify bi-directional TCP connectivity from an ASA to a remote server
  - Inject a simulated TCP SYN packet into an ASA interface
  - ASA processes the injected packet normally and transmits it toward the destination
  - Remote server replies back as it would to the real client
  - ASA processes the response normally and displays the TCP ping result
  - The response packet is discarded by the ASA instead of transmitting to the client
- Easy ASA policy and upstream path verification without client host access
  - TCP RST and ICMP error responses are intercepted and displayed as well

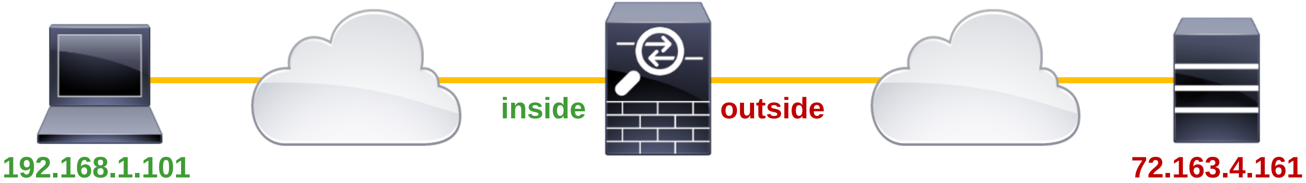


Example: TCP Ping

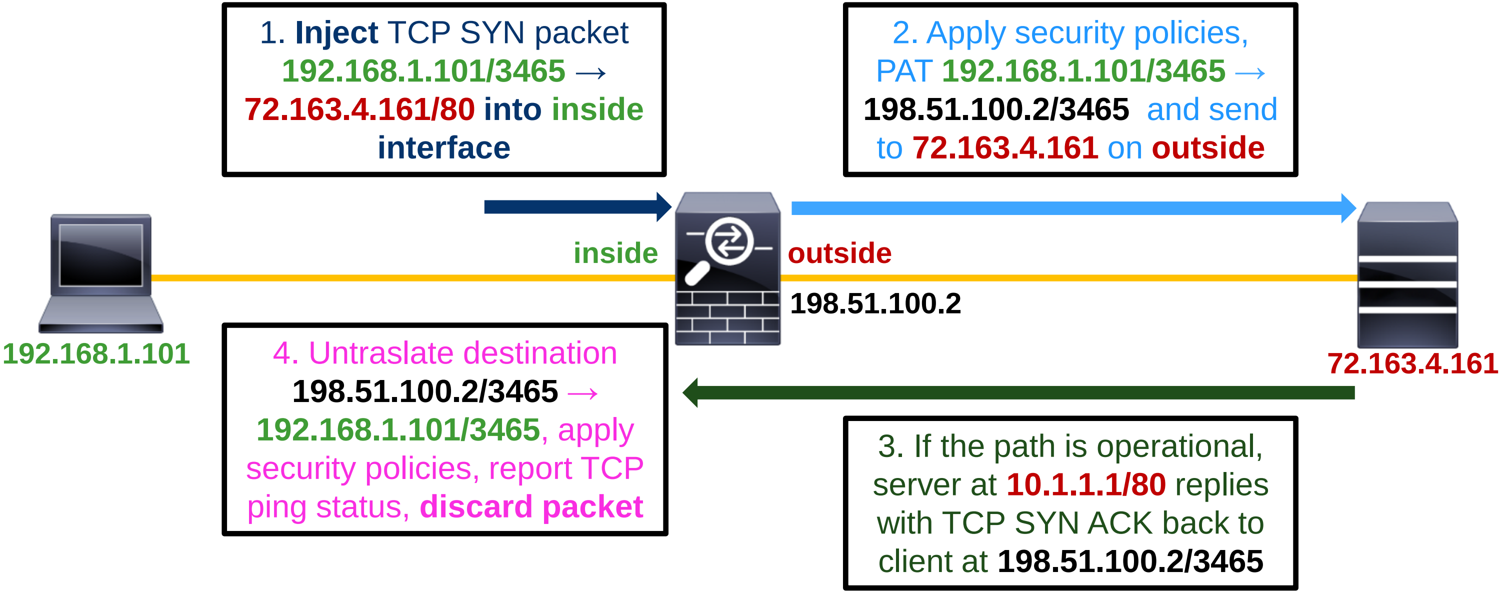
```
asa# ping tcp
 Interface: inside
 Target IP address: 72.163.4.161
 Target IP port: 80
 Specify source? [n]: y
 Source IP address: 192.168.1.101
 Source IP port: [0]
 Repeat count: [5]
 Timeout in seconds: [2]
 Type escape sequence to abort.
 Sending 5 TCP SYN requests to 72.163.4.161 port 80
 from 192.168.1.101 starting port 3465, timeout is 5 seconds:
 !!!!!
 Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

Interface where the test host resides

Real IP address of the test host; the host does not have to be online or even connected



Example: TCP Ping



1. **Inject** TCP SYN packet **192.168.1.101/3465 →** **72.163.4.161/80 into inside interface**

2. Apply security policies, PAT **192.168.1.101/3465 →** **198.51.100.2/3465** and send to **72.163.4.161** on **outside**

3. If the path is operational, server at **10.1.1.1/80** replies with TCP SYN ACK back to client at **198.51.100.2/3465**

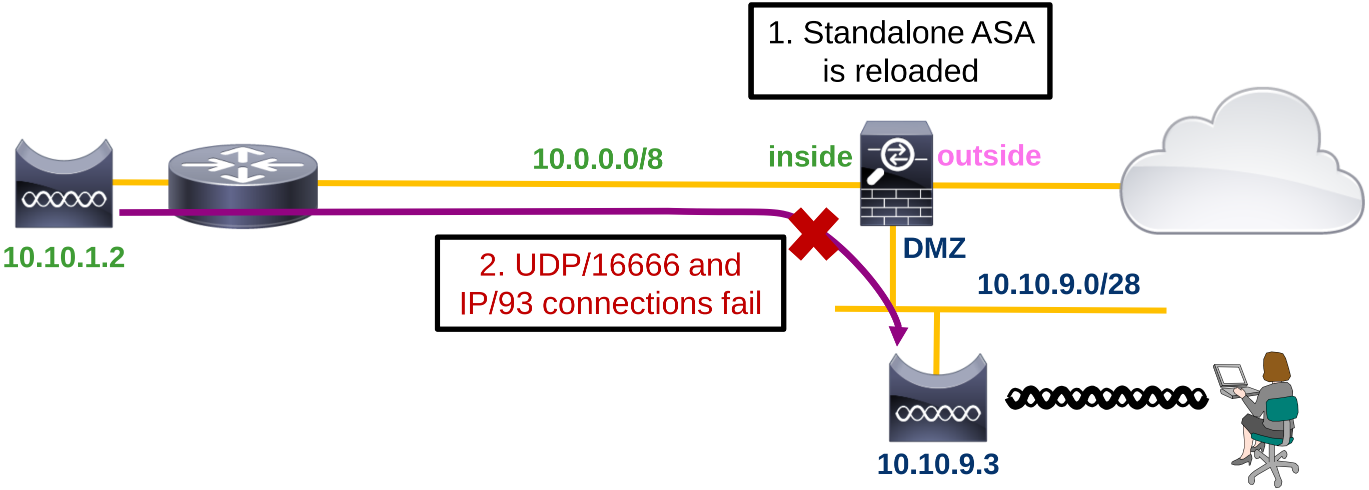
4. Untraslate destination **198.51.100.2/3465 →** **192.168.1.101/3465**, apply security policies, report TCP ping status, **discard packet**

Case Studies

Case Study: UDP Connections Fail After ASA Reload

Problem Summary

- After reloading the ASA, wireless mobility traffic (UDP and IP Protocol 93) from **inside** WLC to **DMZ** WLC fails
- Other traffic (TCP) recovers successfully
- The problem is mitigated by running **clear local-host** on the ASA



1. Standalone ASA is reloaded

2. UDP/16666 and IP/93 connections fail

Checking Connection Table and Drops

- Connections are built and passing traffic through the ASA

```
asa# show conn address 10.10.1.2

126 in use, 12654 most used
97 inside 10.10.9.3 inside 10.10.1.2, idle 0:00:00, bytes 32210
UDP inside 10.10.9.3:16666 inside 10.10.1.2:23124, idle 0:00:00, bytes 4338, flags -
97 inside 10.10.9.3 inside 10.10.1.2, idle 0:00:00, bytes 157240
```

- No packets dropped in ASP and no syslogs of interest

```
asa# capture asp type asp-drop all buffer 1000000
asa# show capture asp | include 10.10.1.2
asa#
asa# show log | include 10.10.1.2
```

Reviewing Packet Captures

Configure separate captures on ingress and egress interfaces

Match the interesting flow bi-directionally

```
asa# capture IN interface inside match udp host 10.10.1.2 host 10.10.9.3
asa# capture OUT interface dmz   match udp host 10.10.1.2 host 10.10.9.3

asa# show capture DMZ
0 packet captured
0 packet shown


asa# show capture IN detail
   1: 19:35:01.371318 0023.0424.ab30 000c.29d7.82ab 10.10.1.2.23124 > 10.10.9.3.16666:  udp 334
   2: 19:35:01.374766 000c.29d7.82ab 0023.0424.ab30 10.10.1.2.23124 > 10.10.9.3.16666:  udp 334
   3: 19:35:02.371128 0023.0424.ab30 000c.29d7.82ab 10.10.1.2.23124 > 10.10.9.3.16666:  udp 334
   4: 19:35:02.374536 000c.29d7.82ab 0023.0424.ab30 10.10.1.2.23124 > 10.10.9.3.16666:  udp 334
```

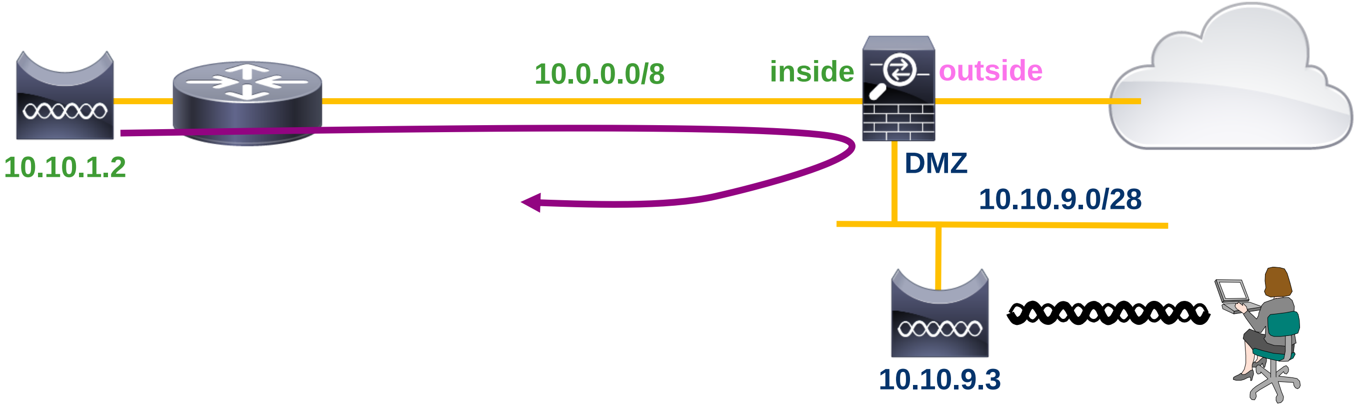
Egress interface capture shows no matching packets

Use detail option to display MAC address information for each frame

Incoming packet from **10.10.1.2** is sent back out of the **inside** interface

U-Turn Connection

- Traffic is looping back out the inside interface back towards the sender



```
asa# sh run | include same-security
same-security-traffic permit intra-interface
```

Allow connections to establish between two endpoints behind the same ASA interface (U-turn)

Checking Packet Tracer

- Packet Tracer shows that a **new** UDP flow will be correctly passed to **DMZ**

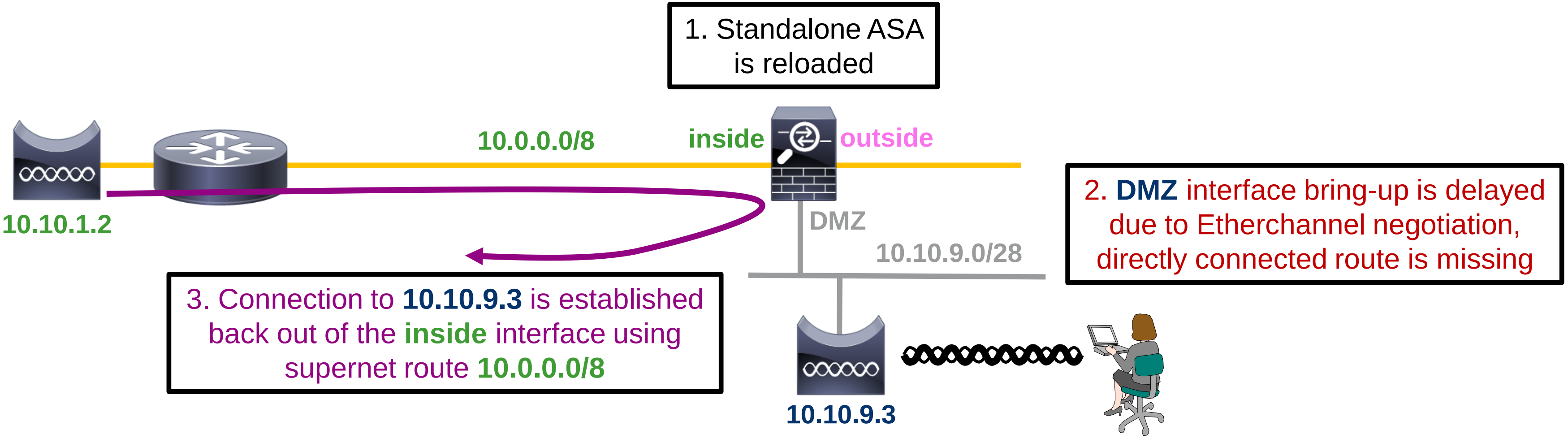
```
asa# packet-tracer input inside udp 10.10.1.22 23124 10.10.9.3 16666
[…]
Phase: 3
Type: ROUTE-LOOKUP
Subtype: input
Result: ALLOW
Config:
Additional Information:
in    10.10.0.0        255.255.0.0      dmz
[…]
Result:
input-interface: inside
input-status: up
input-line-status: up
output-interface: dmz
output-status: up
output-line-status: up
Action: allow
```

Correct routing prefix selected

Flow is allowed

Root Cause

- When conn entry was created, route lookup for **10.10.9.3** resolved to **inside**
- If DMZ interface was not up, the route to **10.10.9.0/28** was not present



1. Standalone ASA is reloaded

2. **DMZ** interface bring-up is delayed due to Etherchannel negotiation, directly connected route is missing

3. Connection to **10.10.9.3** is established back out of the **inside** interface using supernet route **10.0.0.0/8**

Floating Connection Timeout

- The “bad” connection never times out since the UDP traffic is constantly flowing
  - TCP is stateless, so the connection would terminate and re-establish on its own
  - ASA needs to tear the original connection down when the corresponding route changes
  - ASA 8.4(2)+ introduces **timeout floating-conn** to accomplish this goal

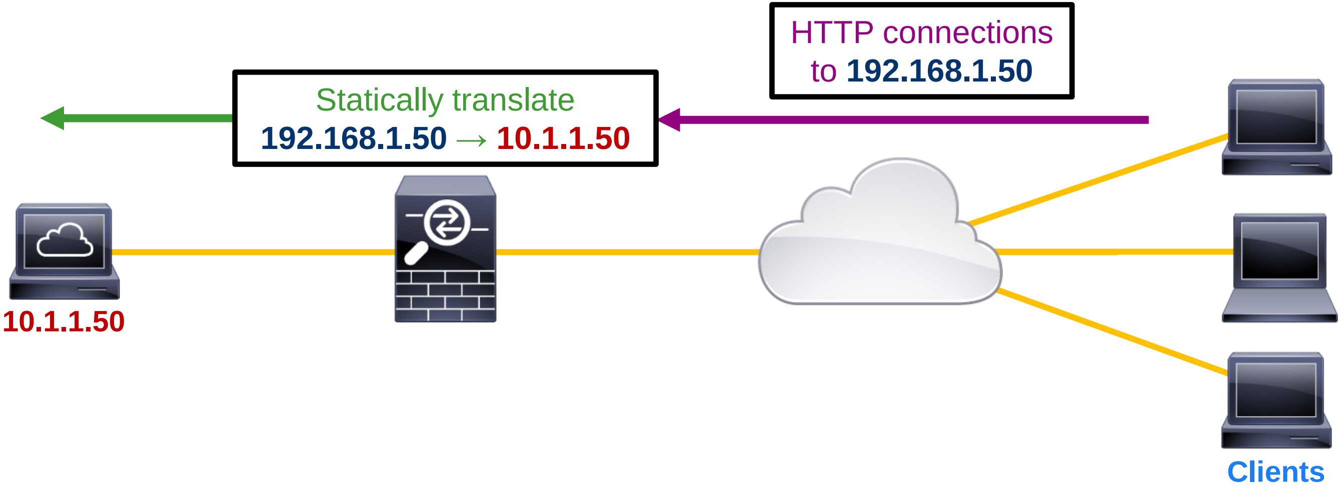
```
asa# show run timeout
timeout xlate 9:00:00
timeout pat-xlate 0:00:30
timeout conn 1:00:00 half-closed 0:10:00 udp 0:02:00 icmp 0:00:02
timeout sunrpc 0:10:00 h323 0:05:00 h225 1:00:00 mgcp 0:05:00 mgcp-pat 0:05:00
timeout sip 0:30:00 sip_media 0:02:00 sip-invite 0:03:00 sip-disconnect 0:02:00
timeout sip-provisional-media 0:02:00 uauth 9:00:00 absolute uauth 0:01:00 inactivity
timeout tcp-proxy-reassembly 0:01:00
timeout floating-conn 0:00:00
asa#
asa# configure terminal
asa(config)# timeout floating-conn 0:01:00
```

Schedule the conn entry for termination in **1 minute** if a matching packet yields a different egress interface on route lookup

Case Study: Intermittent Access to Web Server

Problem Description

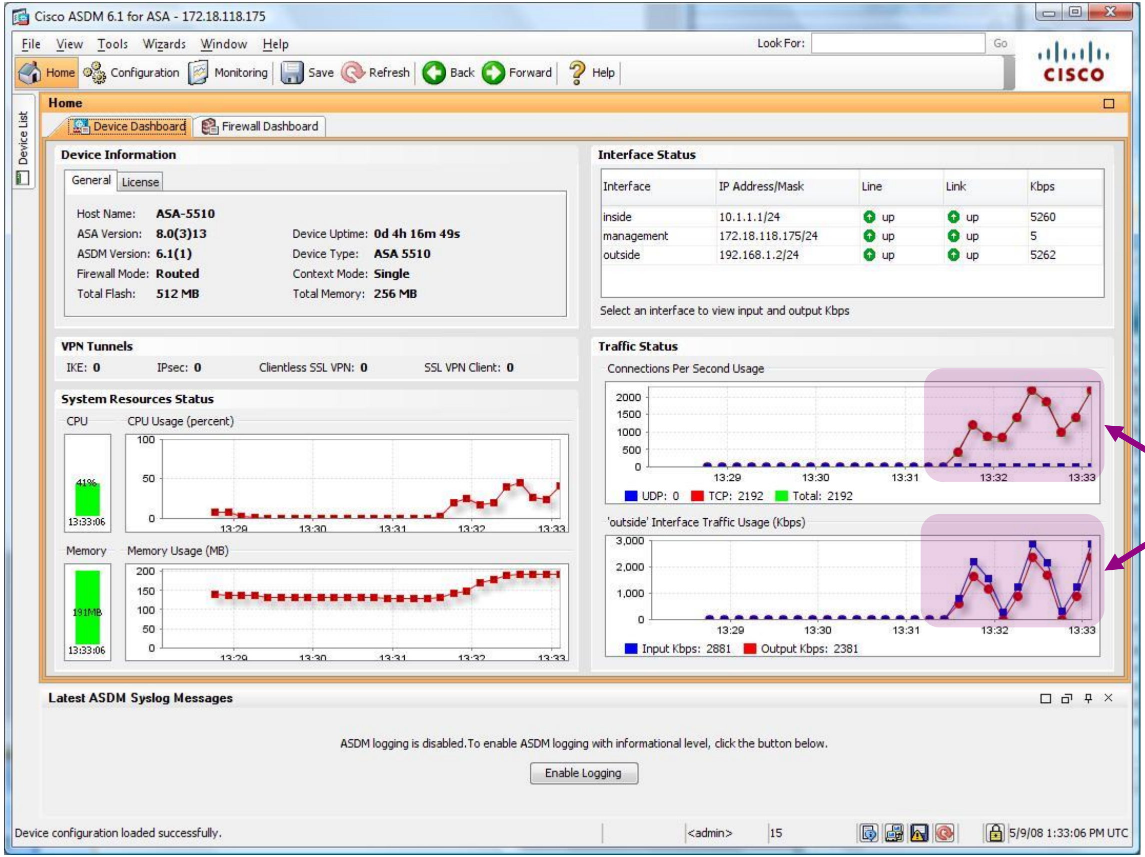
- Public web server is protected by the ASA
- Most external clients are not able to load company's web page



Statically translate 192.168.1.50 → 10.1.1.50

HTTP connections to 192.168.1.50

Monitoring Connection and Traffic Rates in ASDM



Cisco ASDM 6.1 for ASA - 172.18.118.175

File View Tools Wizards Window Help
Look For: Go
Home Configuration Monitoring Save Refresh Back Forward Help
cisco
Home
Device List
Device Dashboard
Firewall Dashboard

Device Information

Host Name: **ASA-5510**
ASA Version: **8.0(3)13**
ASDM Version: **6.1(1)**
Firewall Mode: **Routed**
Total Flash: **512 MB**

Device Uptime: **0d 4h 16m 49s**
Device Type: **ASA 5510**
Context Mode: **Single**
Total Memory: **256 MB**

Interface Status

Interface	IP Address/Mask	Line	Link	Kbps
inside	10.1.1.1/24	up	up	5260
management	172.18.118.175/24	up	up	5
outside	192.168.1.2/24	up	up	5262

Select an interface to view input and output Kbps

VPN Tunnels

IKE: **0** IPsec: **0** Clientless SSL VPN: **0** SSL VPN Client: **0**

Traffic Status

🟦 UDP: 0 🟥 TCP: 2192 🟩 Total: 2192

🟦 Input Kbps: 2881 🟥 Output Kbps: 2381

System Resources Status

Latest ASDM Syslog Messages

ASDM logging is disabled. To enable ASDM logging with informational level, click the button below.

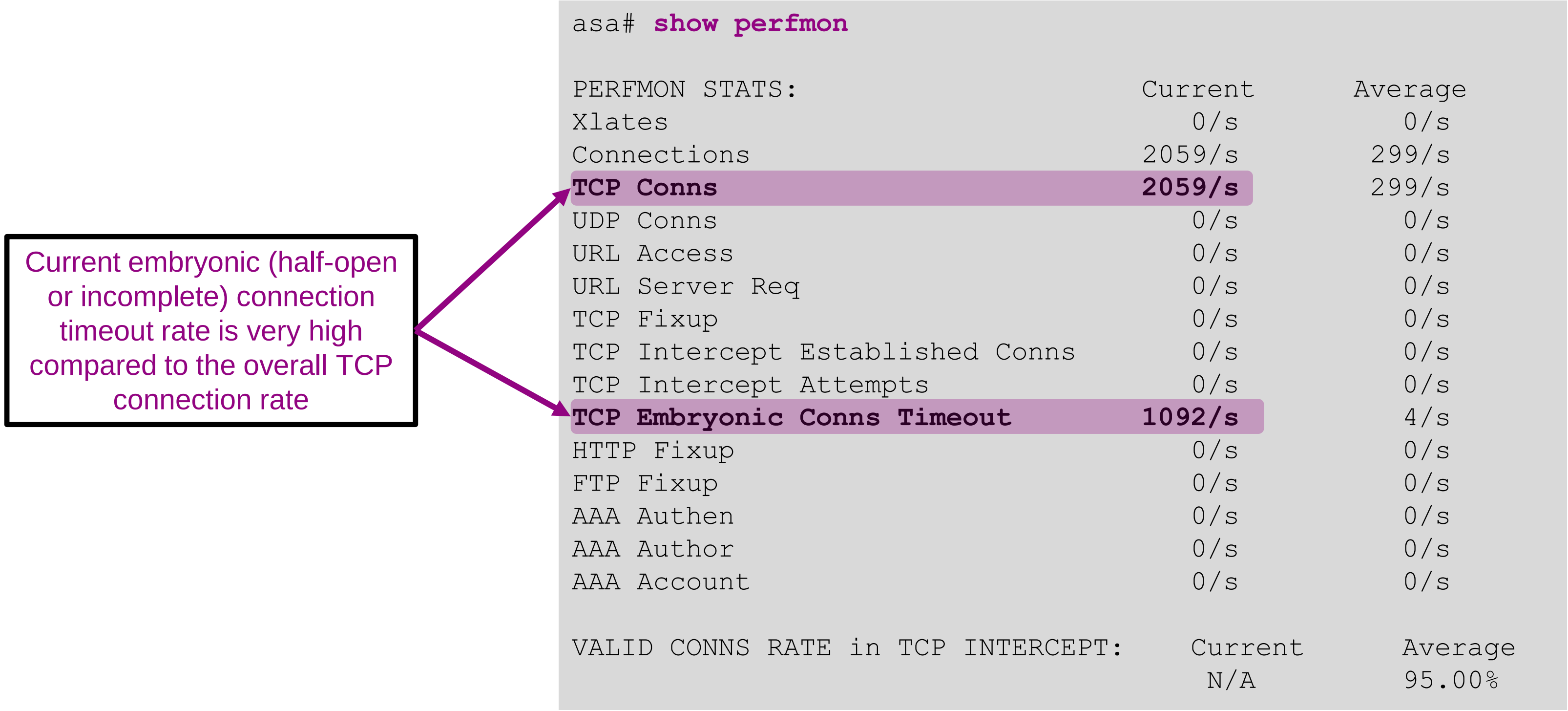
Device configuration loaded successfully.

<admin> 15 5/9/08 1:33:06 PM UTC

Huge connection and traffic spikes on outside interface

Checking Connection Rate Statistics

- **show perfmon** reports xlate, conn, inspection, and AAA transaction rates



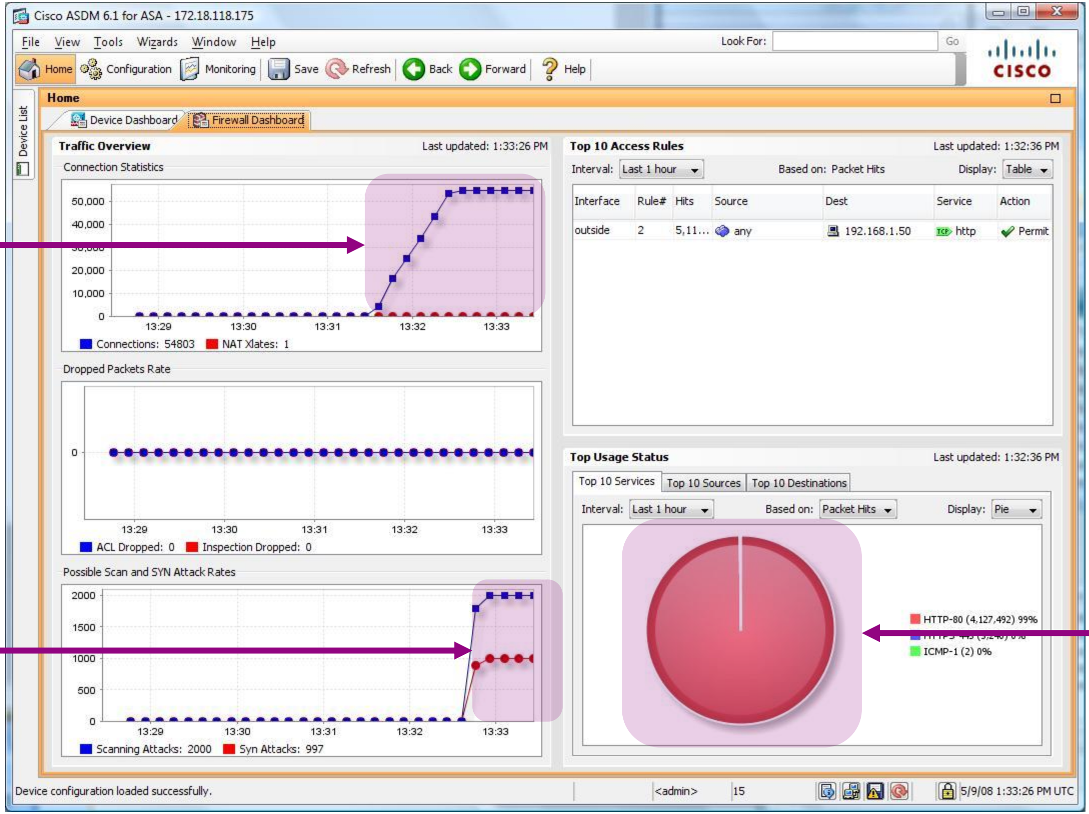
```
asa# show perfmon

PERFMON STATS:                     Current      Average
Xlates                                0/s          0/s
Connections                        2059/s        299/s
TCP Conns                          2059/s        299/s
UDP Conns                             0/s          0/s
URL Access                            0/s          0/s
URL Server Req                        0/s          0/s
TCP Fixup                             0/s          0/s
TCP Intercept Established Conns       0/s          0/s
TCP Intercept Attempts                0/s          0/s
TCP Embryonic Conns Timeout        1092/s          4/s
HTTP Fixup                            0/s          0/s
FTP Fixup                             0/s          0/s
AAA Authen                            0/s          0/s
AAA Author                            0/s          0/s
AAA Account                           0/s          0/s

VALID CONNS RATE in TCP INTERCEPT:    Current      Average
                                       N/A         95.00%
```

Current embryonic (half-open or incomplete) connection timeout rate is very high compared to the overall TCP connection rate

Monitoring SYN Attack Rate in ASDM

Total connection count spikes

High level of incomplete connection attempts indicates a SYN flood attack



Traffic Overview Last updated: 1:33:26 PM

Connection Statistics

🟦 Connections: 54803 🟥 NAT Xlates: 1

Dropped Packets Rate

🟦 ACL Dropped: 0 🟥 Inspection Dropped: 0

Possible Scan and SYN Attack Rates

🟦 Scanning Attacks: 2000 🟥 Syn Attacks: 997

Top 10 Access Rules Last updated: 1:32:36 PM

Interval: Last 1 hour Based on: Packet Hits Display: Table

Interface	Rule#	Hits	Source	Dest	Service	Action
outside	2	5,11...	any	192.168.1.50	TCP http	✓ Permit

Top Usage Status Last updated: 1:32:36 PM

Interval: Last 1 hour Based on: Packet Hits Display: Pie

🟥 HTTP-80 (4,127,492) 99%
🟦 HTTPS-443 (3,240) 0%
🟩 ICMP-1 (2) 0%

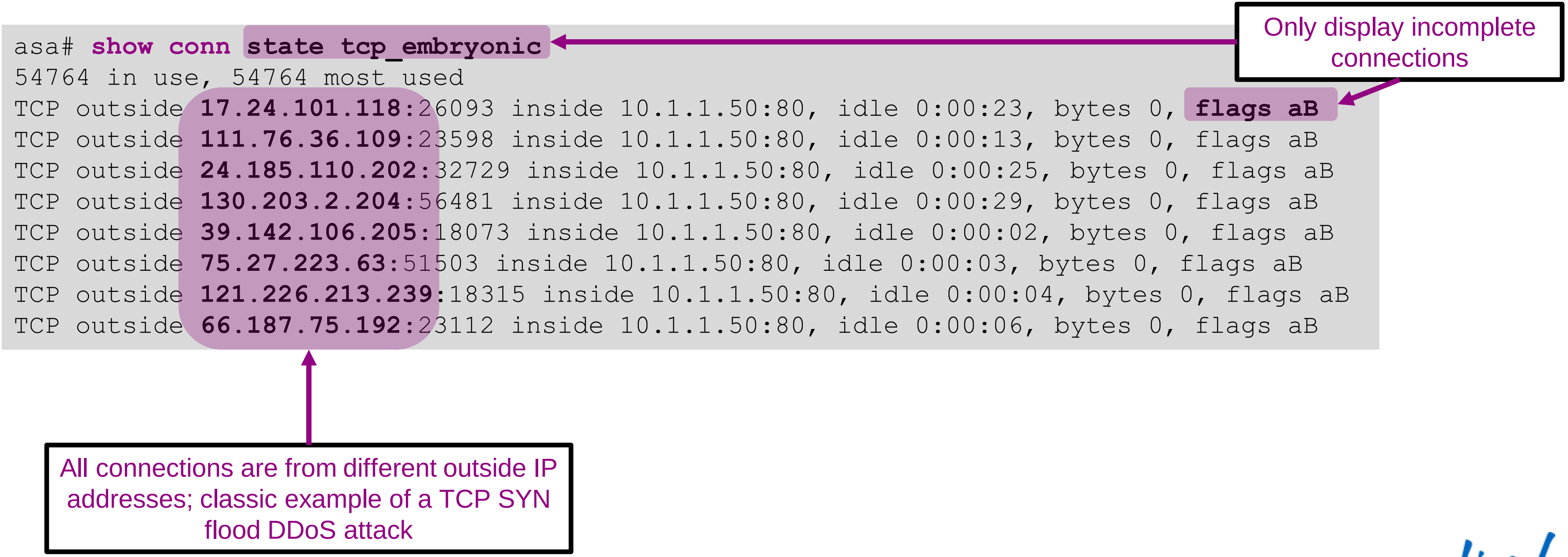
Device configuration loaded successfully.

5/9/08 1:33:26 PM UTC

99% of connections is HTTP

Checking Incomplete TCP Connection Source

- Use **show conn** to see who is creating the incomplete connections



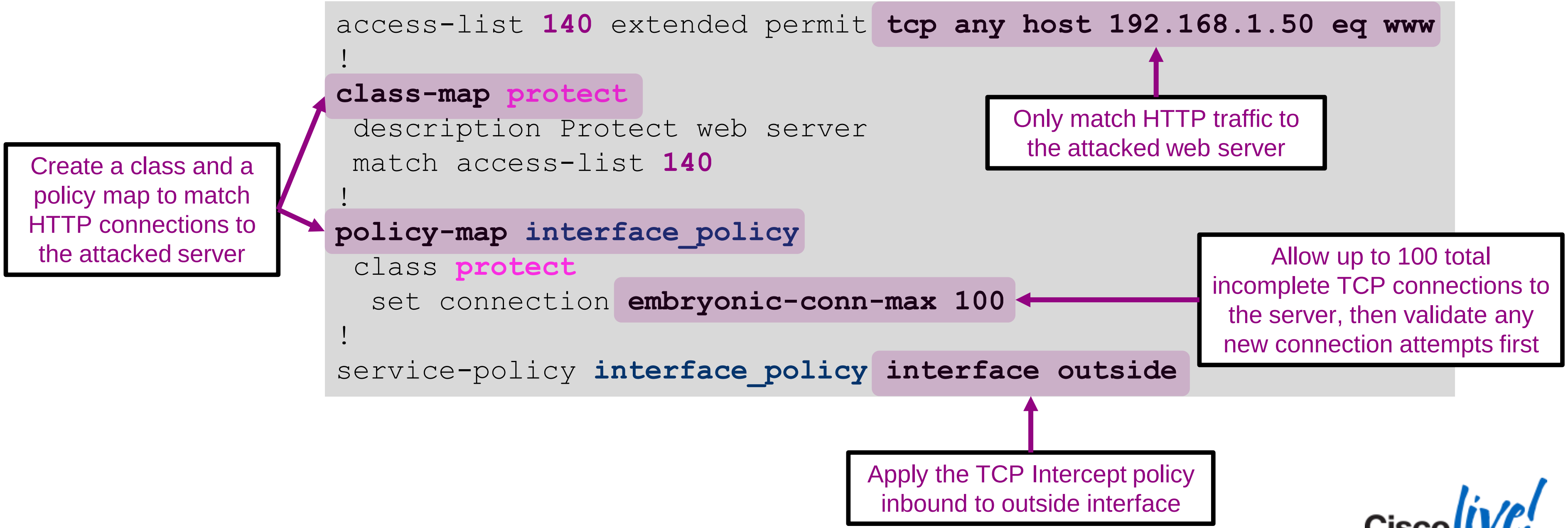
```
asa# show conn state tcp_embryonic
54764 in use, 54764 most used
TCP outside 17.24.101.118:26093 inside 10.1.1.50:80, idle 0:00:23, bytes 0, flags aB
TCP outside 111.76.36.109:23598 inside 10.1.1.50:80, idle 0:00:13, bytes 0, flags aB
TCP outside 24.185.110.202:32729 inside 10.1.1.50:80, idle 0:00:25, bytes 0, flags aB
TCP outside 130.203.2.204:56481 inside 10.1.1.50:80, idle 0:00:29, bytes 0, flags aB
TCP outside 39.142.106.205:18073 inside 10.1.1.50:80, idle 0:00:02, bytes 0, flags aB
TCP outside 75.27.223.63:51503 inside 10.1.1.50:80, idle 0:00:03, bytes 0, flags aB
TCP outside 121.226.213.239:18315 inside 10.1.1.50:80, idle 0:00:04, bytes 0, flags aB
TCP outside 66.187.75.192:23112 inside 10.1.1.50:80, idle 0:00:06, bytes 0, flags aB
```

Only display incomplete connections

All connections are from different outside IP addresses; classic example of a TCP SYN flood DDoS attack

Implementing TCP Intercept

- ASA protects the server from SYN flood by responding with a TCP SYN ACK to validate the client before permitting the connection to the protected server



```
access-list 140 extended permit tcp any host 192.168.1.50 eq www
!
class-map protect
 description Protect web server
 match access-list 140
!
policy-map interface_policy
 class protect
  set connection embryonic-conn-max 100
!
service-policy interface_policy interface outside
```

Only match HTTP traffic to the attacked web server

Create a class and a policy map to match HTTP connections to the attacked server

Allow up to 100 total incomplete TCP connections to the server, then validate any new connection attempts first

Apply the TCP Intercept policy inbound to outside interface

Best Practices

ASA Best Practices

- Avoid interface oversubscription: maximise packet size and minimise rate
- **Baseline** CPU load, connection and xlate counts, and per-interface traffic rates
- **Monitor** vital statistics using MRTG or other SNMP graphing tools
- Selectively apply advanced features to free up CPU
- Record regular **configuration archives** and **show tech** outputs
  - Use Smart Call Home as shown in the Appendix
- Run the latest **maintenance** release in your train to pick up bug fixes
- Upgrade major feature trains **only** for new features or when they mature
  - **Now** is the good time to consider an upgrade to **ASA 9.x** ☺

ASA Best Practices

- **Remove** ACL entries that accumulate 0 hitcount over time
- Log to at least one syslog server, do not configure more than 3
- Move syslog messages you want to see to lower levels or create logging lists instead of raising logging levels and capturing messages you don't want to see
- Use NSEL for recording connection information and **disable** redundant syslogs
- Troubleshoot with syslogs, **show** commands, Packet Tracer, packet captures

Q & A

Complete Your Online Session Evaluation

Give us your feedback and receive a Cisco Live 2014 Polo Shirt!

Complete your Overall Event Survey and 5 Session Evaluations.

- Directly from your mobile device on the Cisco Live Mobile App
- By visiting the Cisco Live Mobile Site www.ciscoliveaustralia.com/mobile
- Visit any Cisco Live Internet Station located throughout the venue

Polo Shirts can be collected in the World of Solutions on Friday 21 March 12:00pm - 2:00pm

Learn online with Cisco Live!

Visit us online after the conference for full access to session videos and presentations. www.CiscoLiveAPAC.com cisco™

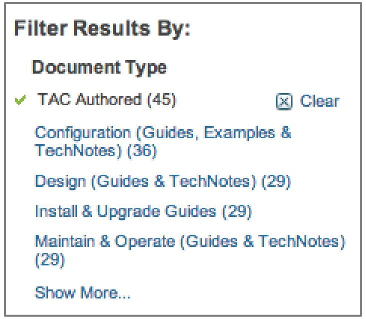
Appendix

Free Online Resources

- Support Communities - Supportforums.cisco.com
- TAC Authored Cisco.com Documents
- TAC Security Show Podcast
- Online learning modules (VoD Training)
- Security RSS Feeds

TAC Authored Cisco.com Content

- Cisco TAC is authoring docs on Cisco.com
- Troubleshooting Guides, Solution Guides, Best Practices, etc



Filter Results By:

Document Type

✓ TAC Authored (45) ☒ Clear

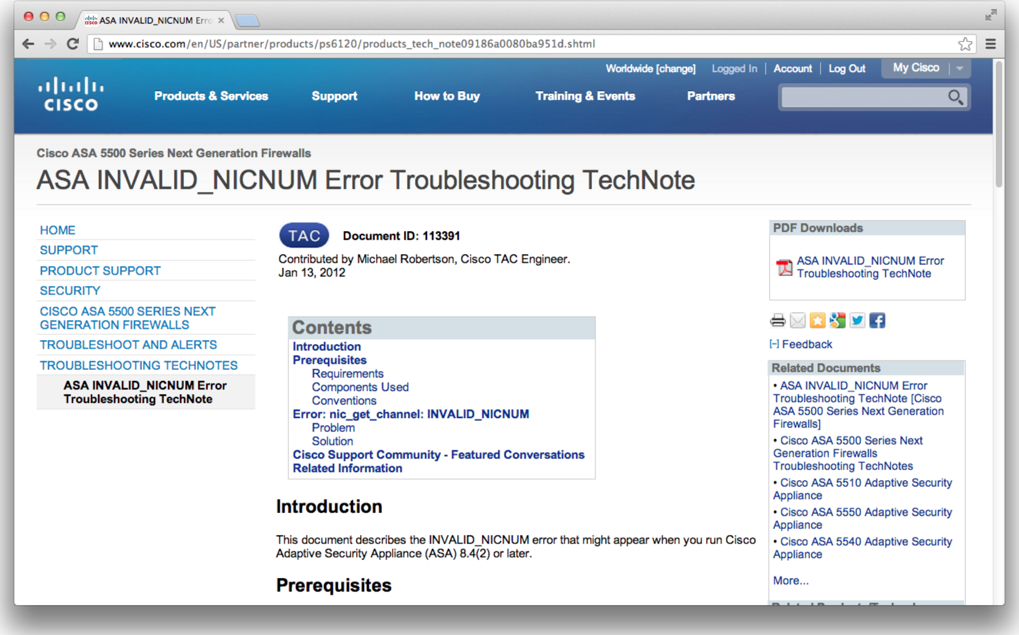
Configuration (Guides, Examples & TechNotes) (36)

Design (Guides & TechNotes) (29)

Install & Upgrade Guides (29)

Maintain & Operate (Guides & TechNotes) (29)

Show More...



Cisco ASA 5500 Series Next Generation Firewalls

ASA INVALID_NICNUM Error Troubleshooting TechNote

HOME
SUPPORT
PRODUCT SUPPORT
SECURITY
CISCO ASA 5500 SERIES NEXT GENERATION FIREWALLS
TROUBLESHOOT AND ALERTS
TROUBLESHOOTING TECHNOTES
ASA INVALID_NICNUM Error Troubleshooting TechNote

TAC **Document ID: 113391**

Contributed by Michael Robertson, Cisco TAC Engineer.
Jan 13, 2012

Contents

Introduction
Prerequisites
  Requirements
  Components Used
  Conventions
Error: nic_get_channel: INVALID_NICNUM
  Problem
  Solution
Cisco Support Community - Featured Conversations
Related Information

Introduction

This document describes the INVALID_NICNUM error that might appear when you run Cisco Adaptive Security Appliance (ASA) 8.4(2) or later.

Prerequisites

PDF Downloads

ASA INVALID_NICNUM Error Troubleshooting TechNote

[-] Feedback

Related Documents

- ASA INVALID_NICNUM Error Troubleshooting TechNote [Cisco ASA 5500 Series Next Generation Firewalls]
- Cisco ASA 5500 Series Next Generation Firewalls Troubleshooting TechNotes
- Cisco ASA 5510 Adaptive Security Appliance
- Cisco ASA 5550 Adaptive Security Appliance
- Cisco ASA 5540 Adaptive Security Appliance

More...

http://supportforums.cisco.com

- Public wiki – anyone can author articles
- Sections for: Firewall, IPS, VPN, and most Cisco technologies
- Hundreds of Sample Configs
- Troubleshooting Docs
- FAQs

Security Hot Issues – RSS Feeds

- Subscribe with an RSS reader
- Receive weekly updates on the Hot Issues customers are facing
- Separate feeds for: ASA, FWSM, ASDM

https://supportforums.cisco.com/docs/DOC-5727

RSS

Redirecting Debugs to Syslog

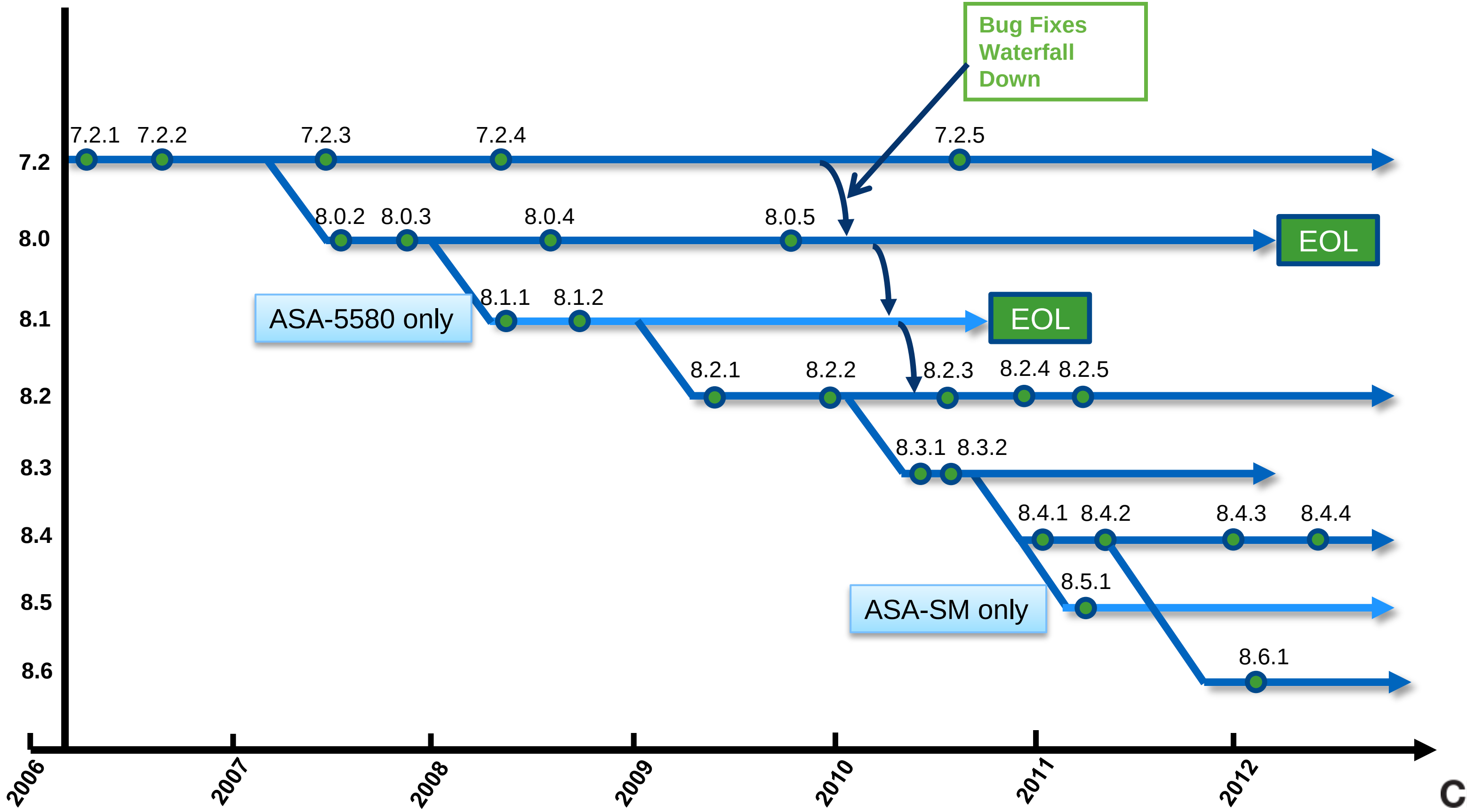
- Problem
  - Log only debug output to syslog
- Solution
  - Create a logging list with only syslog ID 711001
  -
    ```
    ASA(config)# logging list Networkers message 711001
    ```

    Enable debug output to syslogs
  -
    ```
    ASA(config)# logging debug-trace
    INFO: 'logging debug-trace' is enabled. All debug messages
    are currently being redirected to syslog:711001 and will not
    appear in any monitor session
    ```

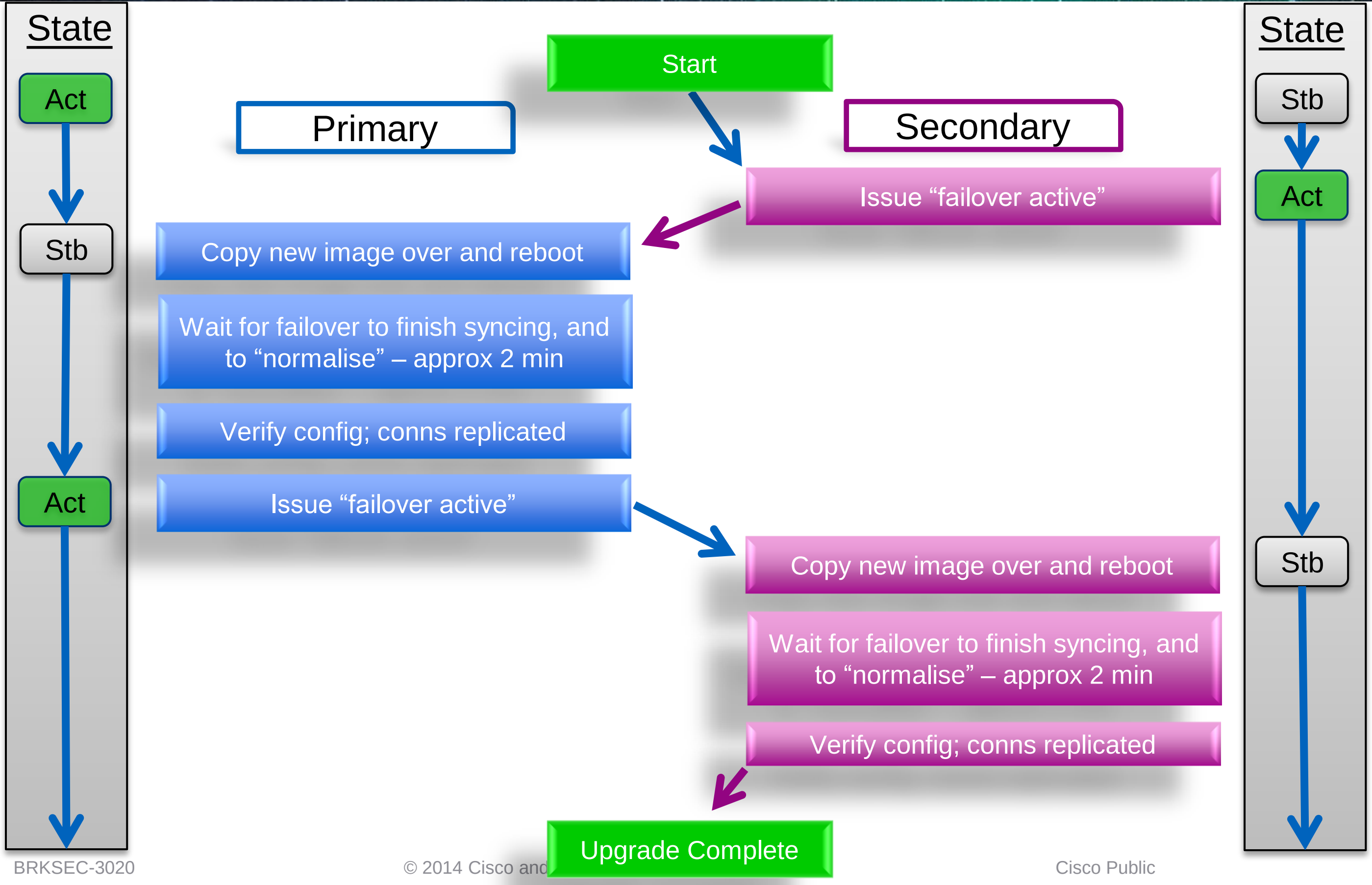
  - Log on the logging list

    ```
    ASA(config)# logging trap Networkers
    ```

ASA Software Trains



High Availability – Zero Downtime Upgrades



State
Act
Stb
Act
Start
Primary
Secondary
Issue “failover active”
Copy new image over and reboot
Wait for failover to finish syncing, and to “normalise” – approx 2 min
Verify config; conns replicated
Issue “failover active”
Copy new image over and reboot
Wait for failover to finish syncing, and to “normalise” – approx 2 min
Verify config; conns replicated
Upgrade Complete
State
Stb
Act
Stb

Failover Interfaces

- Failover Control Link is **vital** to the health of a Failover pair

```
failover lan interface FOVER_CONTROL GigabitEthernet0/0
```

  - Carries bi-directional control, keepalive, and configuration messages
  - Dedicated interface of each unit should connect to an isolated secure network
  - Back-to-back cable connections with a Redundant interface for most protection
  - Failover is **disabled** when Failover Control Link connectivity is interrupted
- Stateful Link latency **should** be <10ms and **must** be <200ms

```
failover link FOVER_STATE GigabitEthernet0/1
```

- Data interface monitoring requires Standby IP addresses
  - Each unit monitors the health of its interfaces and compares with the peer

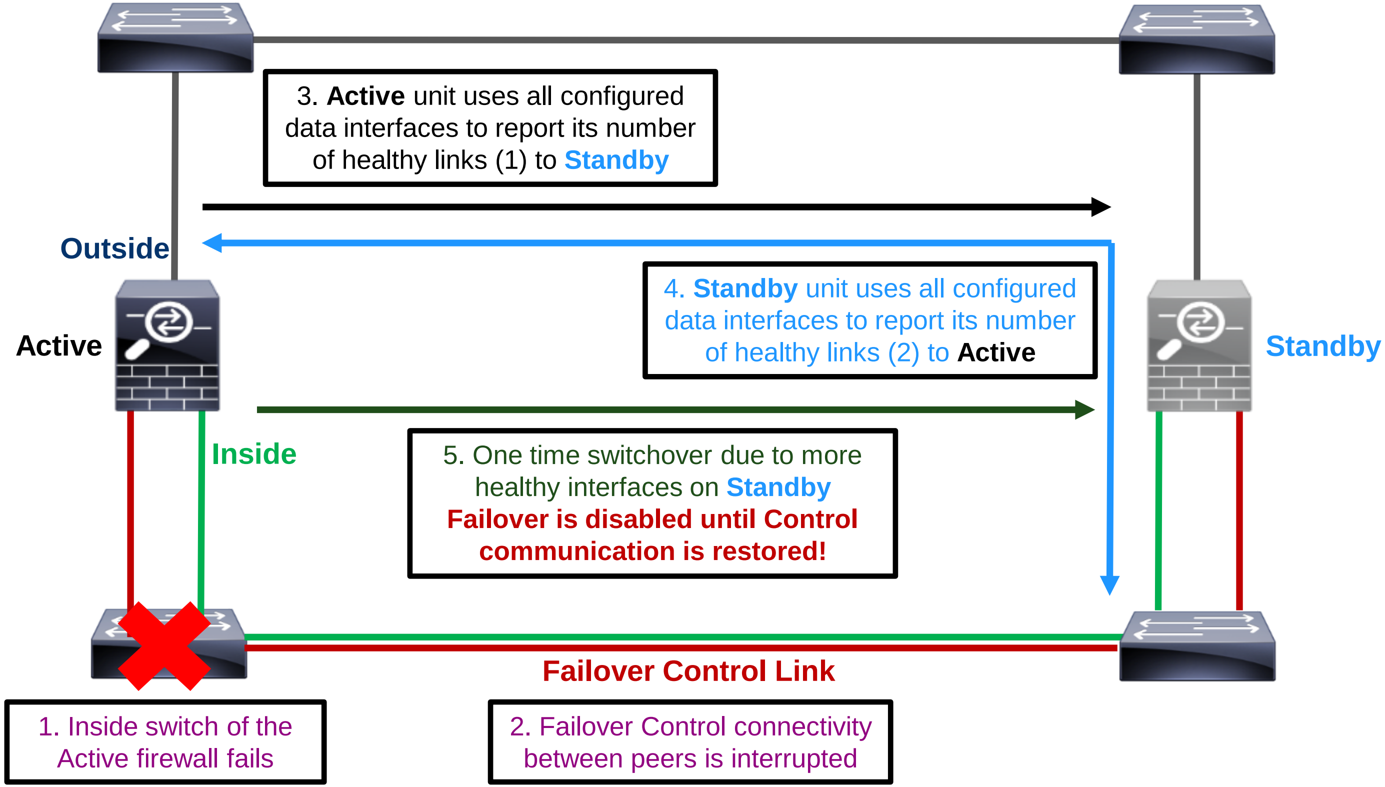
```
ip address 192.168.1.11 255.255.255.0 standby 192.168.1.12
```

  - Active virtual MAC address is inherited from the physical interface of the primary

Failover Health Monitoring

- Local unit monitoring
  - Internal interfaces, expansion cards, service modules
- Failover control link keepalives
- Optional interface monitoring keepalives
  - All physical interfaces by default, but standby IP addresses required
  - Traffic tests when keepalives cease for half the configured holdtime (25 seconds)
  - Interface tests passes with any incoming packets (traffic, ARP, broadcast ping tests)
- **More operationally healthy** unit assumes active role
  - No preemption outside of Active/Active failover

Failover Control Link Failure



3. **Active** unit uses all configured data interfaces to report its number of healthy links (1) to **Standby**

4. **Standby** unit uses all configured data interfaces to report its number of healthy links (2) to **Active**

5. One time switchover due to more healthy interfaces on **Standby**
Failover is disabled until Control communication is restored!

1. Inside switch of the Active firewall fails

2. Failover Control connectivity between peers is interrupted

Quiz: How Well do You Understand Failover?

- What happens when…
  - … you disable failover by issuing **no failover**?
  - … you don’t define standby IP addresses on interfaces?
  - … you replace the primary unit?

Failover Tips and Tricks from TAC

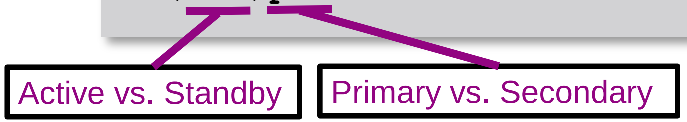
TAC Tip

- Manually configure MAC addresses on all interfaces
- Execute commands on the mate's CLI with **failover exec mate <command>**

```
asa# failover exec mate show memory
Used memory:           31432840 bytes ( 0%)
-------------     ----------------
Total memory:     25769803776 bytes (100%)
```

- Configure the session prompt to indicate failover unit and state

```
asa#
asa(config)# prompt hostname state priority
asa/act/pri(config)# exit
asa/act/pri#
```



Active vs. Standby

Primary vs. Secondary

Clustering Interfaces

- Cluster Control Link carries all communication between cluster members
  - Must use same dedicated interfaces on each member
  - No packet loss or reordering; up to 10ms one-way latency in ASA **9.1(4)+**
  - CCL loss **forces** the member out of the cluster, no back-to-back connections
  - Set MTU 100 bytes above largest data interface MTU
- Mutually elusive data interface modes define external load balancing
- Single virtual IP/MAC across cluster in **Spanned Etherchannel** “**L2**” mode
- Separate IP/MAC on each unit’s data interface in **Individual** “**L3**” mode
- Use only compatible switches
  - Catalyst 3750-X, Catalyst 6500, Nexus 5000, and Nexus 7000 in **9.1(4)+**

Monitoring and Troubleshooting Clustering

- ASDM Clustering dashboard shows aggregated health information
- **show cluster** command group displays aggregated traffic and resource data
  - **show cluster history** helps to understand state transitions and failure reasons
  - **show cluster cpu** helps to check CPU utilisation across cluster
- **show cluster info** command group displays cluster subsystem information
  - **show cluster info health** helps to monitor aggregated unit health data
  - **show cluster info loadbalance** relates to optional Conn Rebalance feature
  - **show cluster info trace** shows cluster state machine debug data for Cisco TAC
- Leverage syslogs to understand failure reasons

```
%ASA-3-747022: Clustering: Asking slave unit terra to quit because it failed interface health
check 3 times (last failure on Port-channel1), rejoin will be attempted after 20 min.
```

  - Use **logging device-id** to identity reporting members for connection events

Example: Show Output Filters

Examples

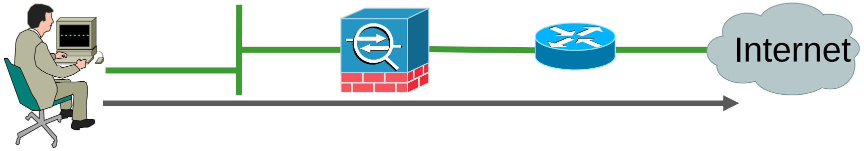
- Display the interface stats starting with the ‘inside’ interface
  - **show interface | begin inside**
- Display the access-list entries that contain address 10.1.1.5
  - **show access-list | grep 10.1.1.5**
- Display the config, except for the access-lists
  - **show run | exclude access-list**
- Display only access-list entries that have non-zero hitcounts
  - **show access-list | grep –v hitcnt=0**
- Display a count of the number of connections each host has
  - **show local-host | include host|count/limit**

```
show <cmd> | begin|include|exclude|grep [-v] <regular_exp>
```

Note: You must Include a Space on Either Side of the Pipe for the Command to Be Accepted; Also, Trailing Spaces Are Counted

Debug ICMP Trace

- Valuable tool used to troubleshoot connectivity issues
- Provides interface and translation information to quickly determine flow
- Echo-replies must be explicitly permitted through ACL, or ICMP inspection must be enabled



Example `debug icmp trace` output

```
ICMP echo-request from inside:10.1.1.2 to 198.133.219.25 ID=3239 seq=4369 length=80
ICMP echo-request: translating inside:10.1.1.2 to outside:209.165.201.22

ICMP echo-reply from outside:198.133.219.25 to 209.165.201.22 ID=3239 seq=4369 length=80
ICMP echo-reply: untranslating outside:209.165.201.22 to inside:10.1.1.2
```

Case Study – Leveraging Smart Call Home

Case Study: Smart Call Home

- **Email ASA command output to you**
- Objective – Send the output of a command directly to your e-mail.
- This is easily accomplished with SCH. Use the command:

  ```
  call-home send <“cmd”> email <email_addr>
  ```

  Example:

  ```
   call-home send “show run” email userid@cisco.com
  ```

- This will send a plain-text e-mail with the output of the command to the e-mail address specified, with the command in the subject line
  - Example:
    Subject: CLI ‘show run’ output

Case Study: Smart Call Home

Collecting Memory Diagnostics over Time

- Objective – Memory appears to be depleting over time on your ASA. Use SCH to collect the detailed memory output hourly, for further investigation.
- This is easily accomplished with SCH. Setting a ”snapshot” **alert-group** to e-mail commands at a specified interval
- Snapshot will contain the following command:

```
show conn count
show memory detail
```

Case Study: Smart Call Home

Example Config

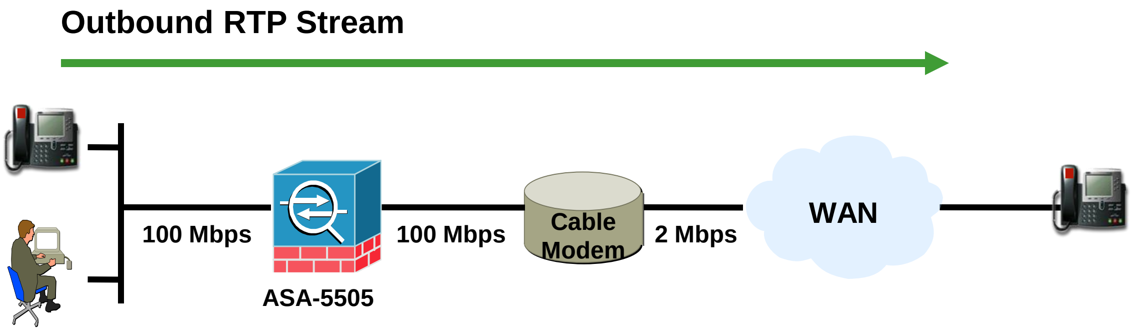
```
service call-home
call-home
 alert-group-config snapshot
  add-command “show conn count”
  add-command “show memory detail”
 contact-email-addr user@cisco.com
 sender from user@cisco.com
 sender reply-to user@cisco.com
 mail-server smtp-server.cisco.com priority 1
 profile SENDCMD
  active
  destination address email user@cisco.com
  destination preferred-msg-format long-text
  destination transport-method email
  subscribe-to-alert-group snapshot periodic hourly
```

Case Study: Poor Voice Quality

Case Study: Poor Voice Quality

Problem

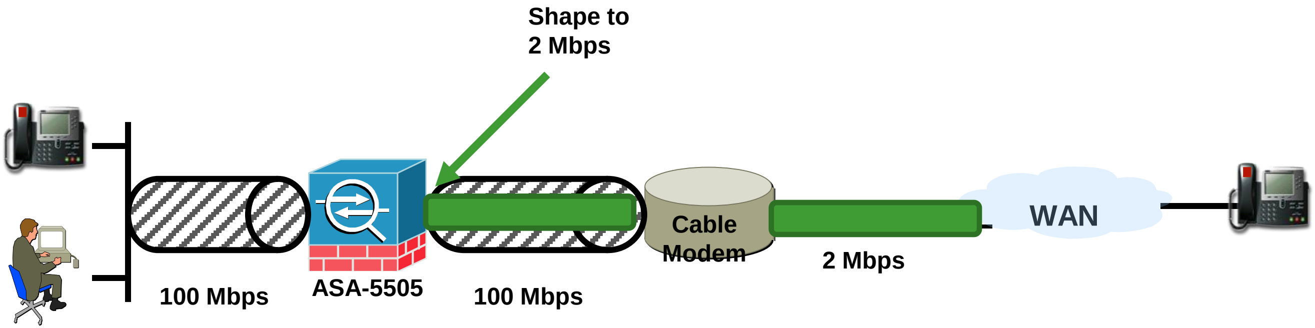
- Poor outbound voice quality at SOHO sites



Case Study: Poor Voice Quality

Solution: Traffic Shaping

- What is traffic shaping, and why is it needed here?
- Why won’t policing work?
- Why won’t priority queuing alone work?



Case Study: Poor Voice Quality – Configuration Example (Traffic Shaping)

Solution

- Prioritise voice traffic and shape all traffic down to 2 Mbps on the outside interface.

```
class-map voice-traffic
 match dscp af13 ef
!
policy-map qos_class_policy
 class voice-traffic
  priority
!
policy-map qos_outside_policy
 class class-default
  shape average 2000000
  service-policy qos_class_policy
!
service-policy qos_outside_policy interface outside
```

- To view statistics on the operation of the shaper, use the command `show service-policy shape`

Case Study: Poor Voice Quality

Things to Keep in Mind:

- Shaping can only be applied to the class **class-default**
- Shaping only works in the outbound direction on an interface
- The shaping value is in bits per second, and must be a multiple of 8000
- The shaping policy is applied to all sub-interfaces on a physical interface
- Not supported on the ASA-5580 platform
- Not supported in Transparent or Multi-context mode

Show Process cpu-hog

- The `show processes cpu-hog` command displays a list of processes, and the function stack (Traceback) which executed, and lead to a process running on the CPU longer than the minimum platform threshold

```
ASA# show processes cpu-hog
Process:       ssh_init, NUMHOG: 18, MAXHOG: 15, LASTHOG: 10
LASTHOG At:    14:18:47 EDT May 29 2009
PC:            8b9ac8c (suspend)
Traceback:     8b9ac8c  8ba77ed  8ba573e  8ba58e8  8ba6971
               8ba02b4  8062413

CPU hog threshold (msec): 10.240
Last cleared: None
```

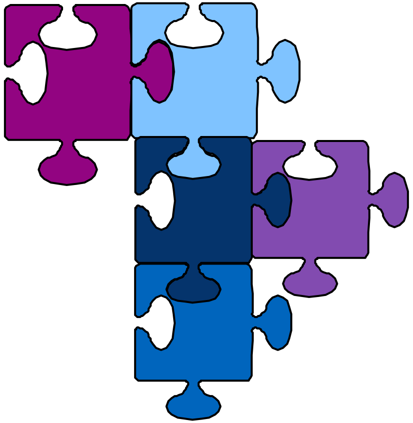
  - A corresponding syslog message is also generated
    Note: The Traceback syslog below does not signify a crash

```
May 29 2009 14:18:47: %ASA-7-711002: Task ran for 10 msec,
Process = ssh_init, PC = 8b9ac8c, Traceback =   0x08B9AC8C  0x08BA77ED
0x08BA573E  0x08BA58E8  0x08BA6971  0x08BA02B4  0x08062413
```

Case Study – Advanced Syslog Analysis

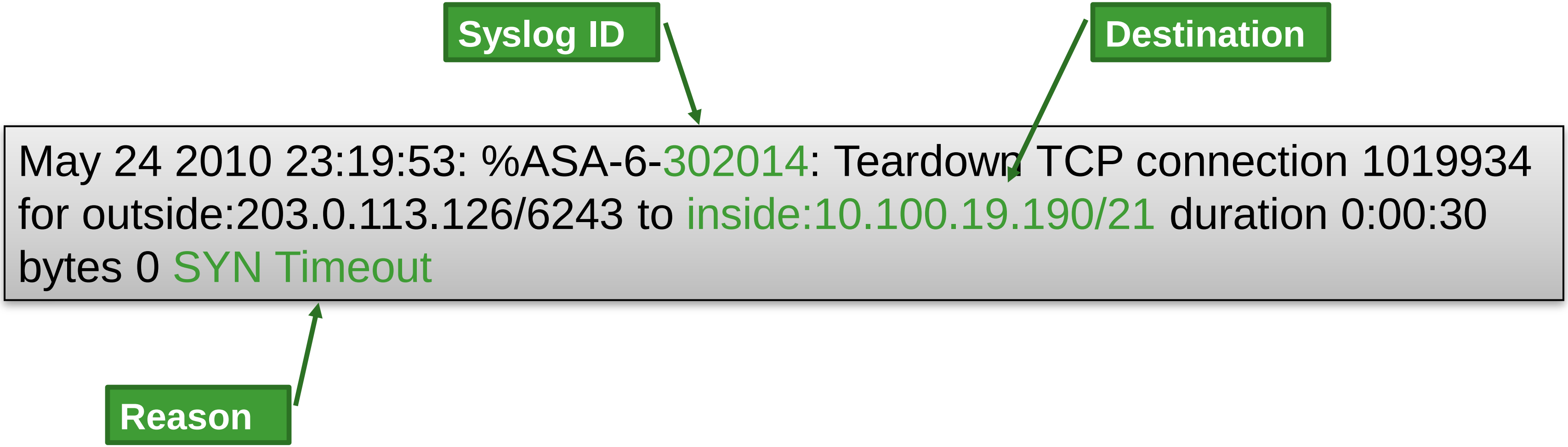
Case Study: Advanced Syslog Analysis

- Problem – Find Services which are permitted through the firewall, yet the servers no longer exist
- Get a fast Linux/Solaris machine with a decent amount of memory
- Learn to use the following commands:
  - cat
  - grep, egrep, fgrep
  - cut
  - awk (basic)
  - sort
  - uniq
  - Perl (advanced manipulation)
- Pipe the commands to construct the necessary outputs!



Case Study: Advanced Syslog Analysis

- Interesting syslogs appear as follows:



```
May 24 2010 23:19:53: %ASA-6-302014: Teardown TCP connection 1019934
for outside:203.0.113.126/6243 to inside:10.100.19.190/21 duration 0:00:30
bytes 0 SYN Timeout
```

Syslog ID

Destination

Reason

Case Study: Advanced Syslog Analysis

Results:

- grep – used to find the syslogs we want
- awk – used to print the destination column (IP/port)
- uniq – used to print only unique entries, with a count
- sort – used to display ordered list, highest count first

```
syslogserver-sun% grep 302014 syslog.txt | grep "SYN Timeout" | awk '{print $13}' | uniq
-c | sort -r -n

 673  inside:10.100.19.190/21
 451  dmz:192.168.5.13/80
 392  dmz:192.168.5.11/443
 358  inside:10.0.0.67/1521
 119  inside:10.0.1.142/80
```

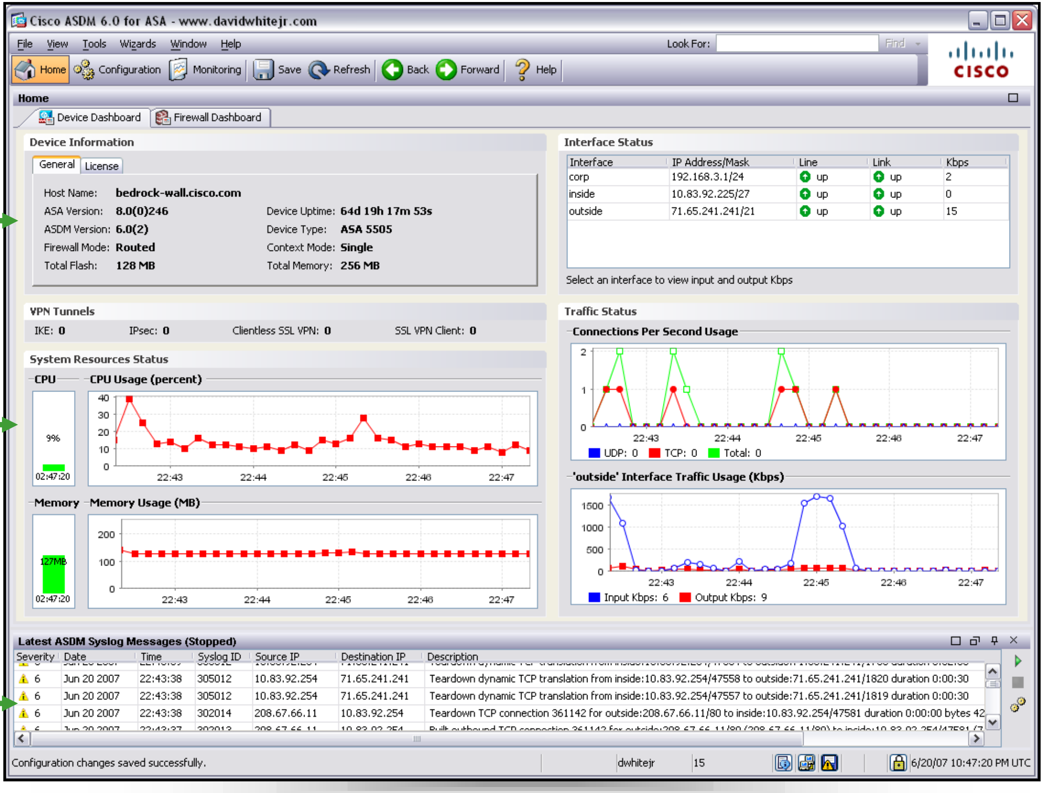
ASDM

ASDM Home Page

Device Information

CPU, Memory, Conns/Sec, Interface Traffic

Real-Time Syslogs



Cisco ASDM 6.0 for ASA - www.davidwhitejr.com
File View Tools Wizards Window Help
Look For: Find
Home Configuration Monitoring Save Refresh Back Forward Help
CISCO
Home
Device Dashboard
Firewall Dashboard

Device Information

General | License

Host Name:	bedrock-wall.cisco.com		
ASA Version:	8.0(0)246	Device Uptime:	64d 19h 17m 53s
ASDM Version:	6.0(2)	Device Type:	ASA 5505
Firewall Mode:	Routed	Context Mode:	Single
Total Flash:	128 MB	Total Memory:	256 MB

Interface Status

Interface	IP Address/Mask	Line	Link	Kbps
corp	192.168.3.1/24	up	up	2
inside	10.83.92.225/27	up	up	0
outside	71.65.241.241/21	up	up	15

Select an interface to view input and output Kbps

VPN Tunnels

IKE: 0 IPsec: 0 Clientless SSL VPN: 0 SSL VPN Client: 0

Traffic Status

System Resources Status

Latest ASDM Syslog Messages (Stopped)

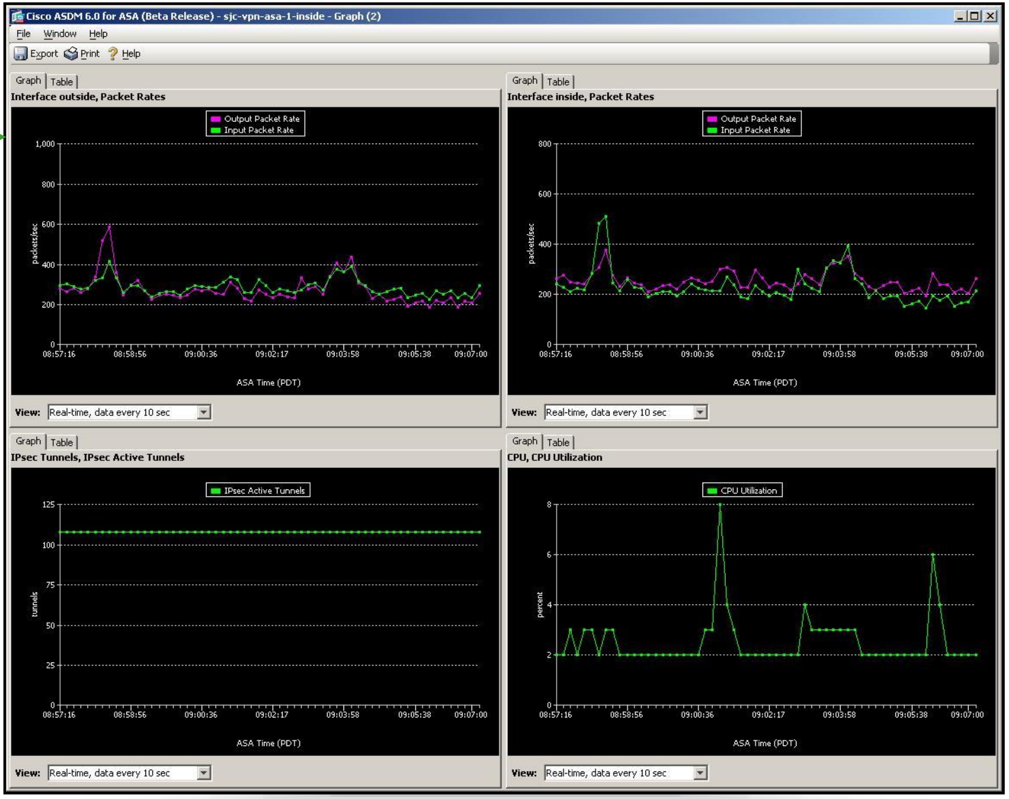
Severity	Date	Time	Syslog ID	Source IP	Destination IP	Description
6	Jun 20 2007	22:43:38	305012	10.83.92.254	71.65.241.241	Teardown dynamic TCP translation from inside:10.83.92.254/47558 to outside:71.65.241.241/1820 duration 0:00:30
6	Jun 20 2007	22:43:38	305012	10.83.92.254	71.65.241.241	Teardown dynamic TCP translation from inside:10.83.92.254/47557 to outside:71.65.241.241/1819 duration 0:00:30
6	Jun 20 2007	22:43:38	302014	208.67.66.11	10.83.92.254	Teardown TCP connection 361142 for outside:208.67.66.11/80 to inside:10.83.92.254/47581 duration 0:00:00 bytes 42
6	Jun 20 2007	22:43:37	302013	208.67.66.11	10.83.92.254	Built outbound TCP connection 361142 for outside:208.67.66.11/80 (208.67.66.11/80) to inside:10.83.92.254/47581 (7

Configuration changes saved successfully. dwhitejr 15 6/20/07 10:47:20 PM UTC

Using ASDM for Monitoring

Great for Monitoring Trends

Up to Four Different Graphs Can Be Displayed



View: Real-time, data every 10 sec

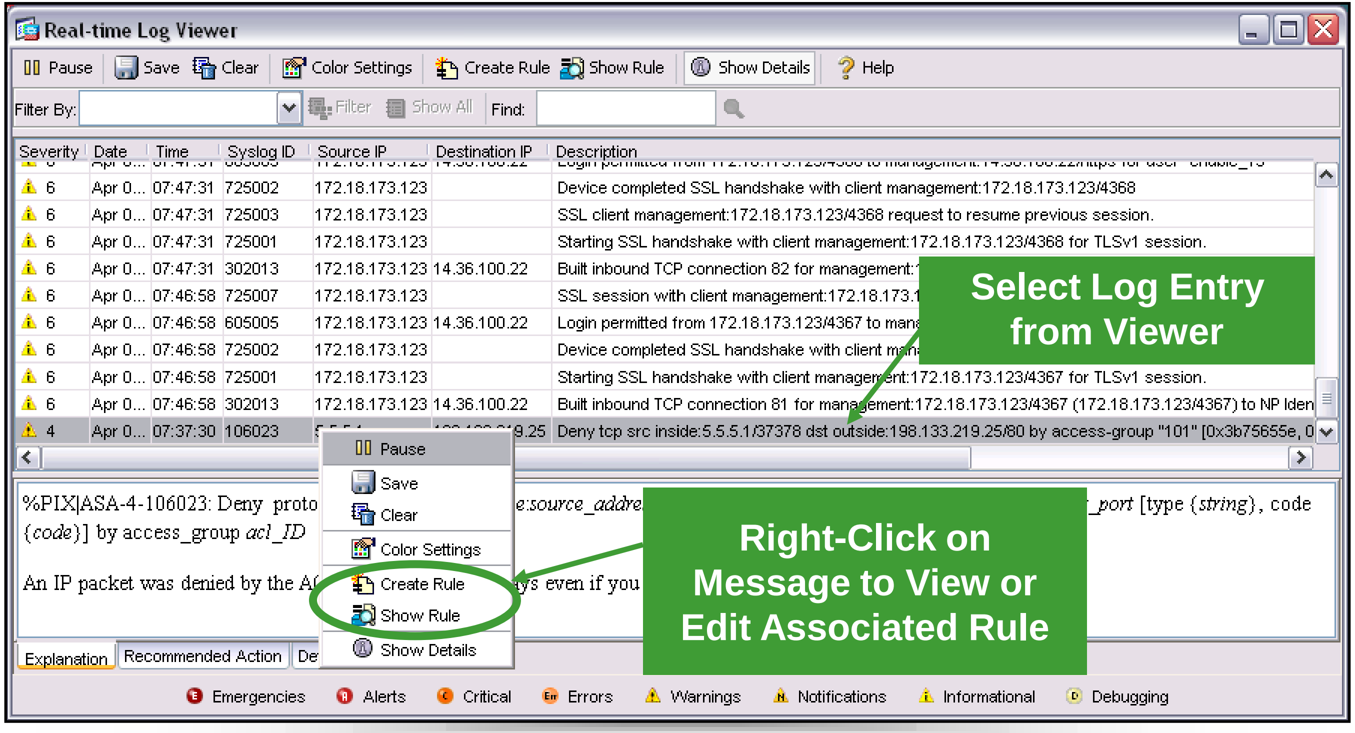
View: Real-time, data every 10 sec

View: Real-time, data every 10 sec

View: Real-time, data every 10 sec

ASDM

Editing Rules from the Log Viewer



Severity	Date	Time	Syslog ID	Source IP	Destination IP	Description
6	Apr 0...	07:47:31	605005	172.18.173.123	14.36.100.22	Login permitted from 172.18.173.123/4368 to management:14.36.100.22/https for user "enable_15"
6	Apr 0...	07:47:31	725002	172.18.173.123		Device completed SSL handshake with client management:172.18.173.123/4368
6	Apr 0...	07:47:31	725003	172.18.173.123		SSL client management:172.18.173.123/4368 request to resume previous session.
6	Apr 0...	07:47:31	725001	172.18.173.123		Starting SSL handshake with client management:172.18.173.123/4368 for TLSv1 session.
6	Apr 0...	07:47:31	302013	172.18.173.123	14.36.100.22	Built inbound TCP connection 82 for management:1
6	Apr 0...	07:46:58	725007	172.18.173.123		SSL session with client management:172.18.173.1
6	Apr 0...	07:46:58	605005	172.18.173.123	14.36.100.22	Login permitted from 172.18.173.123/4367 to mana
6	Apr 0...	07:46:58	725002	172.18.173.123		Device completed SSL handshake with client mana
6	Apr 0...	07:46:58	725001	172.18.173.123		Starting SSL handshake with client management:172.18.173.123/4367 for TLSv1 session.
6	Apr 0...	07:46:58	302013	172.18.173.123	14.36.100.22	Built inbound TCP connection 81 for management:172.18.173.123/4367 (172.18.173.123/4367) to NP Iden
4	Apr 0...	07:37:30	106023	5.5.5.1	9.25	Deny tcp src inside:5.5.5.1/37378 dst outside:198.133.219.25/80 by access-group "101" [0x3b75655e, 0

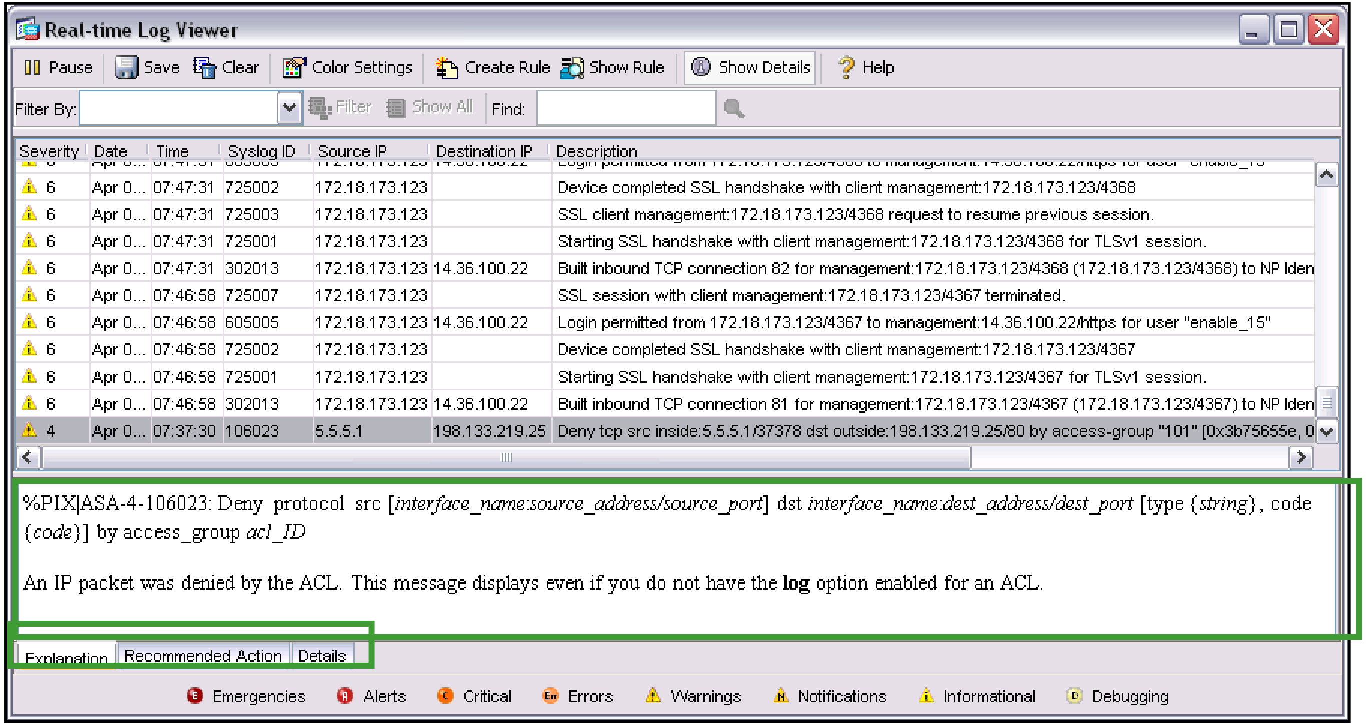
%PIX|ASA-4-106023: Deny proto e:source_addre _port [type {*string*}, code {*code*}] by access_group *acl_ID*

An IP packet was denied by the A ys even if you

Select Log Entry from Viewer

Right-Click on Message to View or Edit Associated Rule

ASDM: Syslogs Explained



Severity	Date	Time	Syslog ID	Source IP	Destination IP	Description
6	Apr 0...	07:47:31	605005	172.18.173.123	14.36.100.22	Login permitted from 172.18.173.123/4368 to management:14.36.100.22/https for user "enable_15"
6	Apr 0...	07:47:31	725002	172.18.173.123		Device completed SSL handshake with client management:172.18.173.123/4368
6	Apr 0...	07:47:31	725003	172.18.173.123		SSL client management:172.18.173.123/4368 request to resume previous session.
6	Apr 0...	07:47:31	725001	172.18.173.123		Starting SSL handshake with client management:172.18.173.123/4368 for TLSv1 session.
6	Apr 0...	07:47:31	302013	172.18.173.123	14.36.100.22	Built inbound TCP connection 82 for management:172.18.173.123/4368 (172.18.173.123/4368) to NP Iden
6	Apr 0...	07:46:58	725007	172.18.173.123		SSL session with client management:172.18.173.123/4367 terminated.
6	Apr 0...	07:46:58	605005	172.18.173.123	14.36.100.22	Login permitted from 172.18.173.123/4367 to management:14.36.100.22/https for user "enable_15"
6	Apr 0...	07:46:58	725002	172.18.173.123		Device completed SSL handshake with client management:172.18.173.123/4367
6	Apr 0...	07:46:58	725001	172.18.173.123		Starting SSL handshake with client management:172.18.173.123/4367 for TLSv1 session.
6	Apr 0...	07:46:58	302013	172.18.173.123	14.36.100.22	Built inbound TCP connection 81 for management:172.18.173.123/4367 (172.18.173.123/4367) to NP Iden
4	Apr 0...	07:37:30	106023	5.5.5.1	198.133.219.25	Deny tcp src inside:5.5.5.1/37378 dst outside:198.133.219.25/80 by access-group "101" [0x3b75655e, 0

%PIX|ASA-4-106023: Deny protocol src [*interface_name*:*source_address*/*source_port*] dst *interface_name*:*dest_address*/*dest_port* [type {*string*}, code {*code*}] by access_group *acl_ID*

An IP packet was denied by the ACL. This message displays even if you do not have the **log** option enabled for an ACL.

cisco
TM