
TOMORROW
starts here.
CISCO
Cisco live!

Troubleshooting ASA Firewalls

BRKSEC-3020

Prapanch Ramamoorthy – Engineer, Cisco Services

#clmel

Cisco live!

Your Speaker

Prapanch Ramamoorthy
Engineer, Technical Services

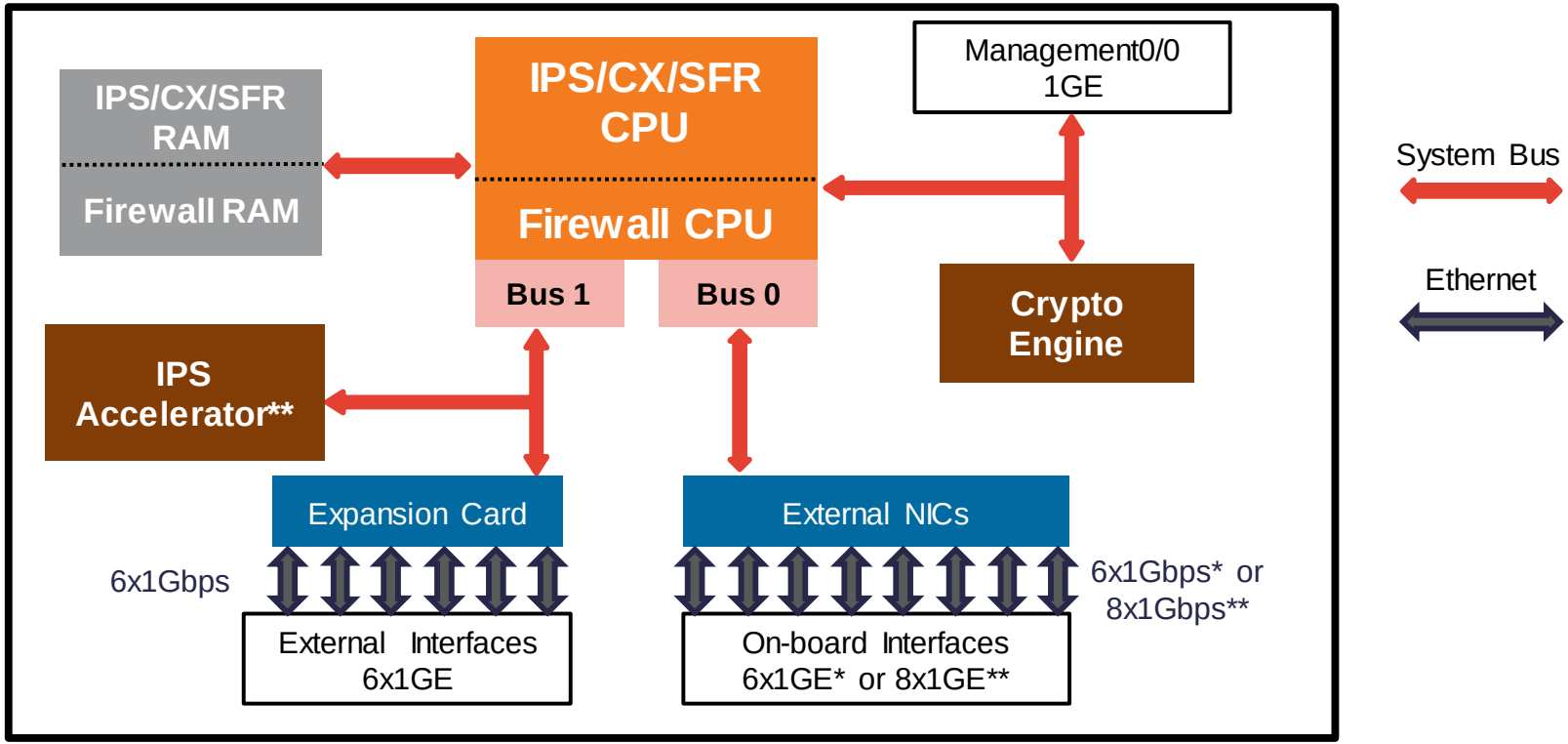
6 years of TAC experience
Primarily focused on security

Agenda

- Architecture
- Packet Flow
- Troubleshooting Tools
- Case Studies
- Best Practices

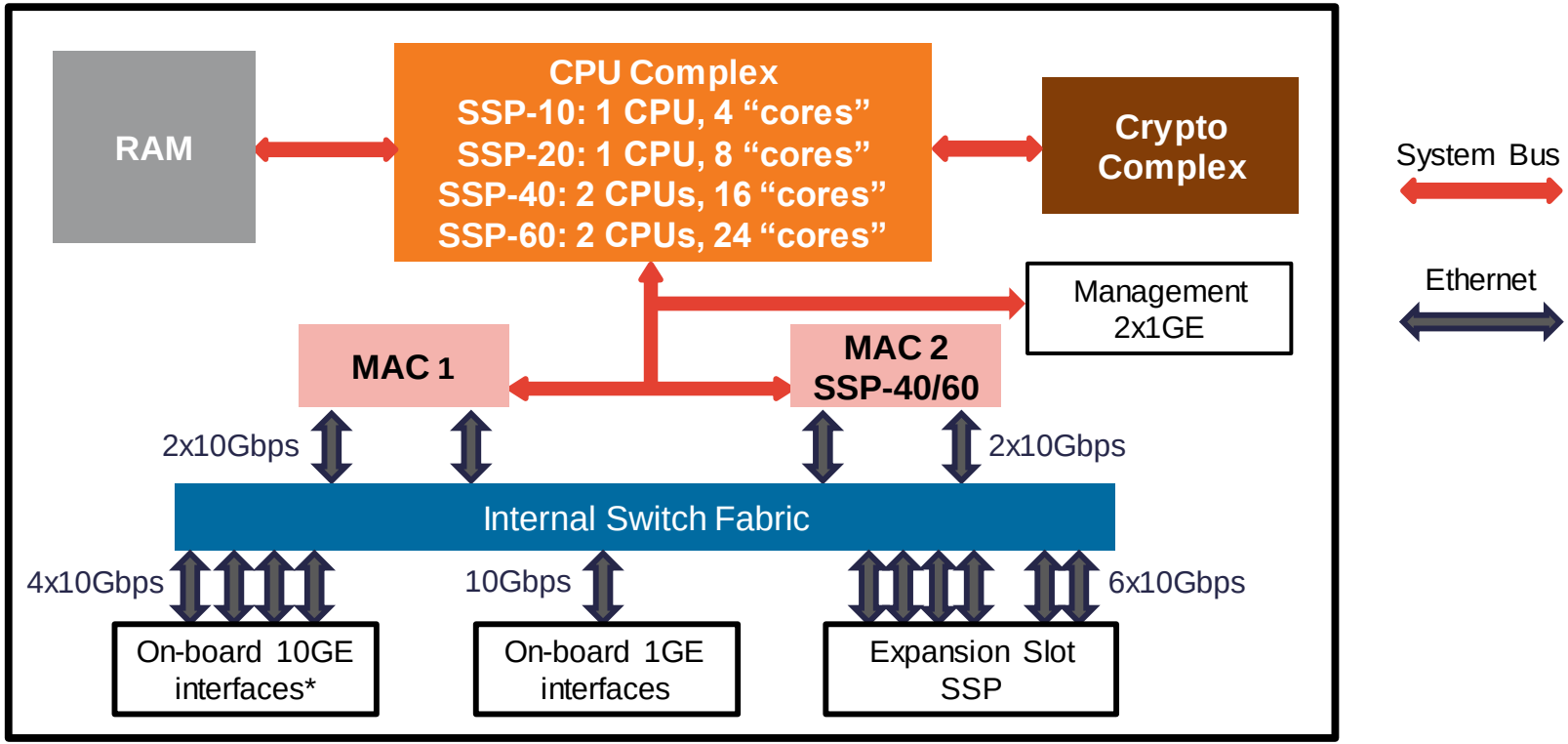
Architecture

ASA 5500-X Block Diagram



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

ASA 5585-X Block Diagram

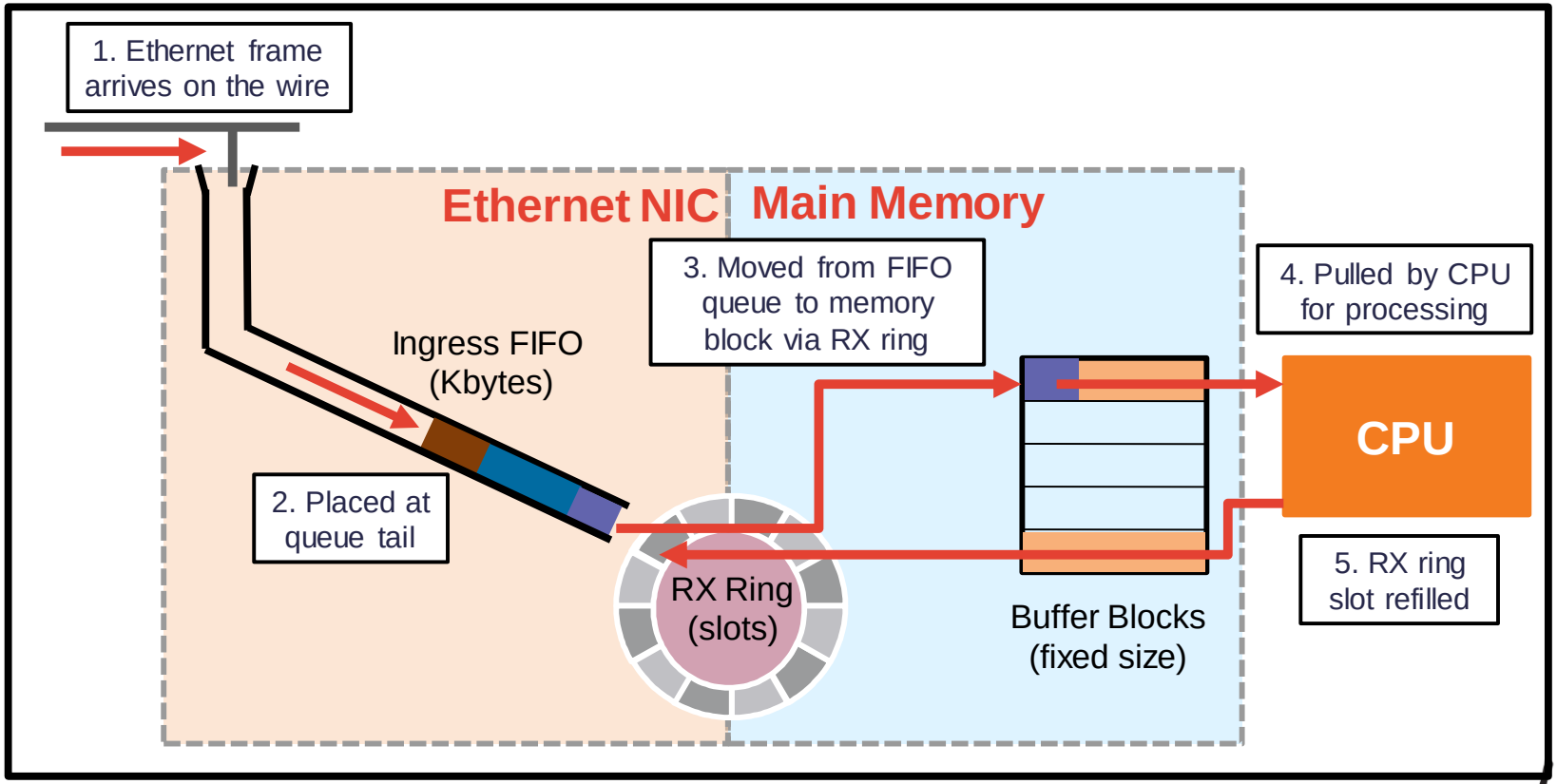


*2 on SSP-10/20 and 4 on SSP-40/60

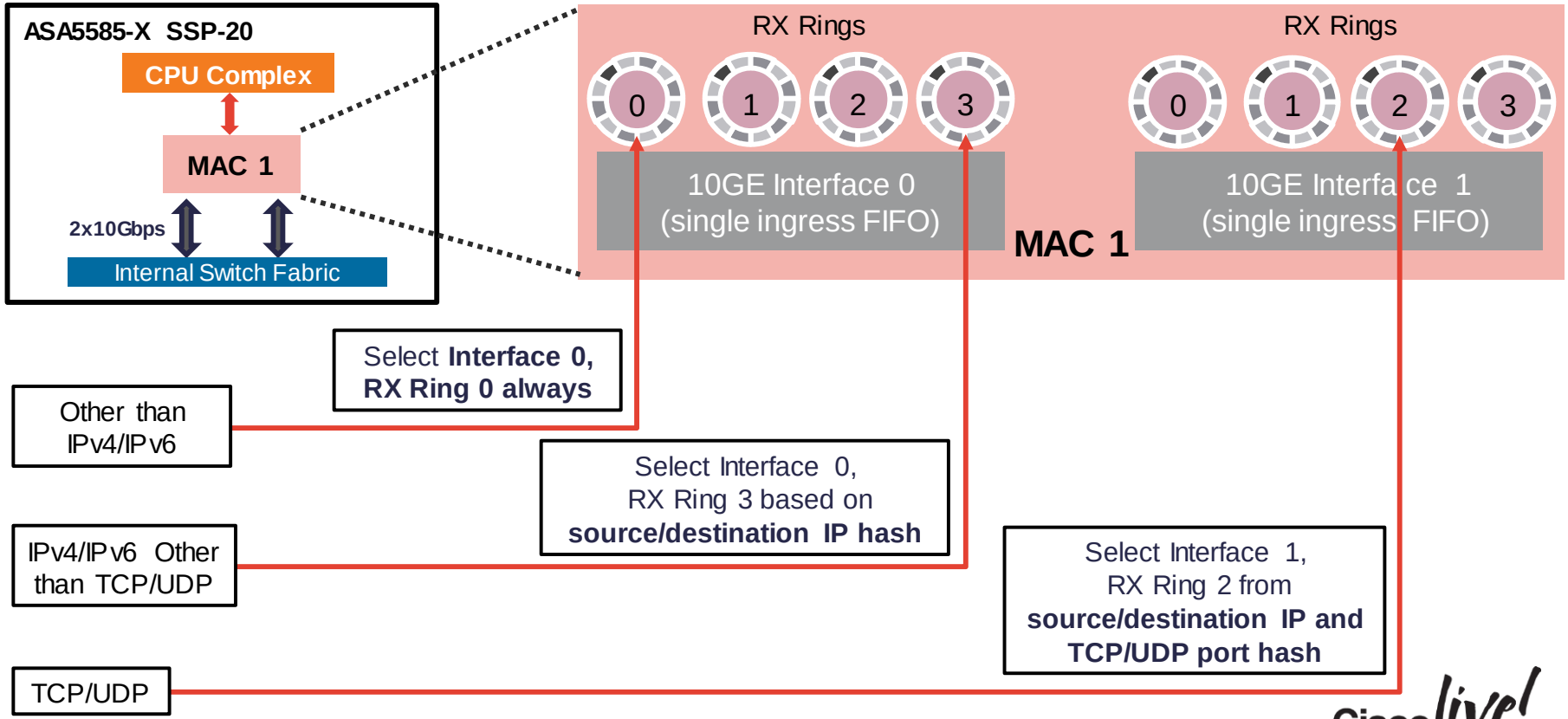
Ingress Frame Processing

- 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

Ingress Frame Processing



Ingress Load-Balancing on 10GE and MAC



10GE MAC Interface Information

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

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

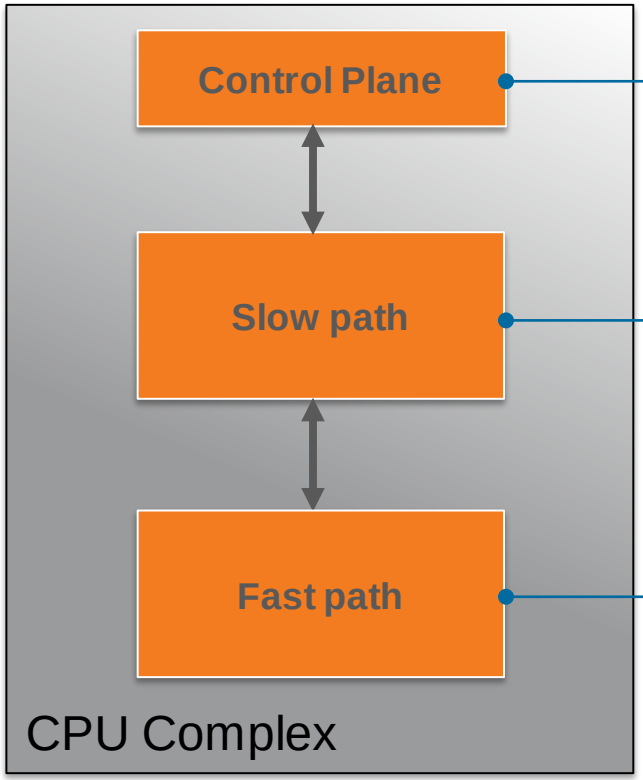
```
asa# show interface detail | begin Internal-Data
Interface Internal-Data0/0 "", is up, line protocol is up
  Hardware is i82599_xaui rev01, BW 10000 Mbps, DLY 10 usec
[…]
        Queue Stats:
          RX[00]: 325778 packets, 31260705 bytes, 0 overrun
                  Blocks free curr/low: 511/509
          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
           […]
```

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



Control plane

Application inspection, management traffic, failover, clustering, etc.

Slow path

New conn creation, ACL check, NAT check, logging

Fast Path

Processes existing connections, find egress interface, NAT, transmit packet

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

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
<snip>

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            6509062          0  6374429   134633  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           0
lacp               728568          0   728568        0   728568           0
```

Request queue

Requests in queue

Max requests ever in queue

Individual event

Allocation attempts

No memory

Blocks put into queue

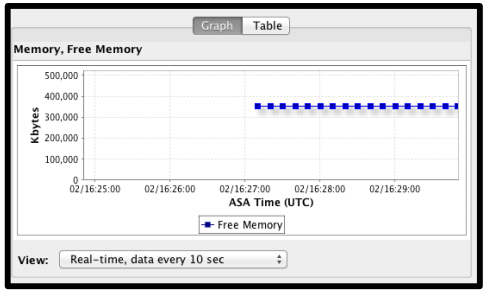
Times queue limit reached

```
%ASA-4-447001: ASA DP to CP ARP Event Queue was full. Queue length 2048, limit 2048
```

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 caching

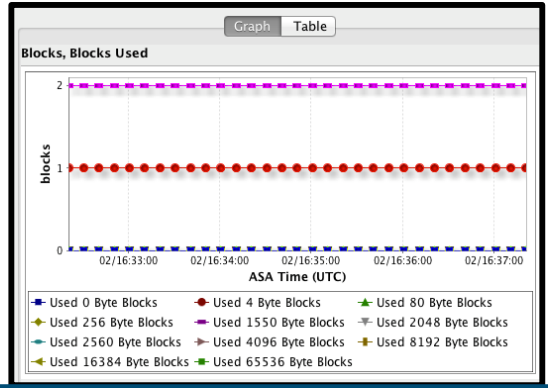
Cisco ASA — Memory Blocks

Number of blocks allocated at bootup

Current number of free blocks available

```
ASA# show blocks
  SIZE    MAX    LOW    CNT
     0    400    397    400
     4    100     99     99
    80    403    379    401
   256   1200   1190   1195
  1550   6511    803    903
  2048   1200   1197   1200
  2560    264    264    264
  4096    100    100    100
  8192    100    100    100
 16384    102    102    102
 65536     16     16     16
ASA#
```

Least number of free blocks since bootup



Use ASDM to graph free blocks available

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

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**

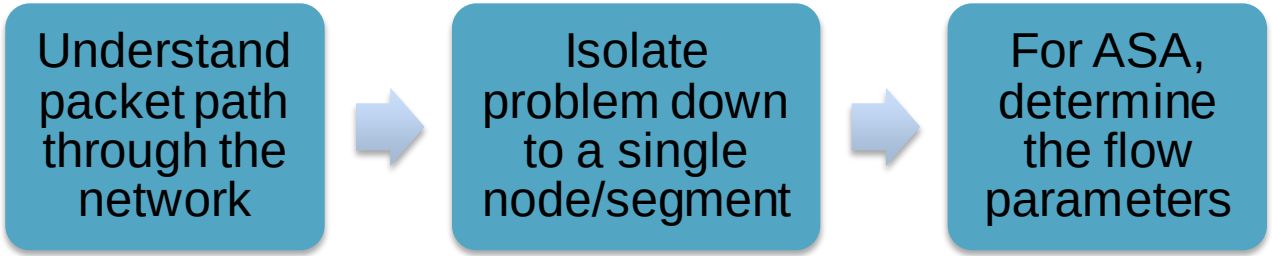
- Nested Object Groups magnify the impact
  - Add a new source Object Group with 25 additional objects
  - Result: (10+25) x 21 x 33 = **24,255 rules** (ACEs)
- ACL Optimisation prevents the Object Group expansion
  - Significant reduction in memory utilisation, not so much on CPU

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

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

Packet Flow

Troubleshooting Methodology



For problems relating to the Cisco ASA, ask these questions:

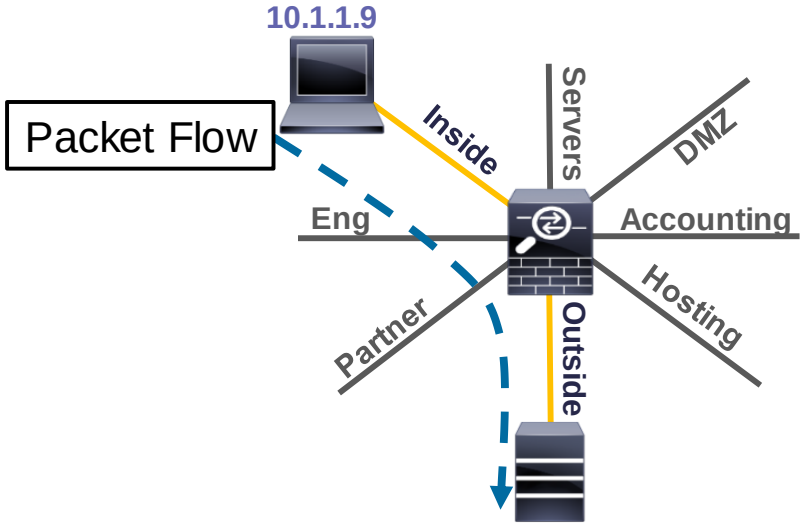
- What is the Protocol? – TCP/UDP/GRE, etc.
- What are the Source and Destination IP addresses?
- What are the Source and Destination ports (if applicable)?
- What are the logical interfaces (named) associated with the flow?

```
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

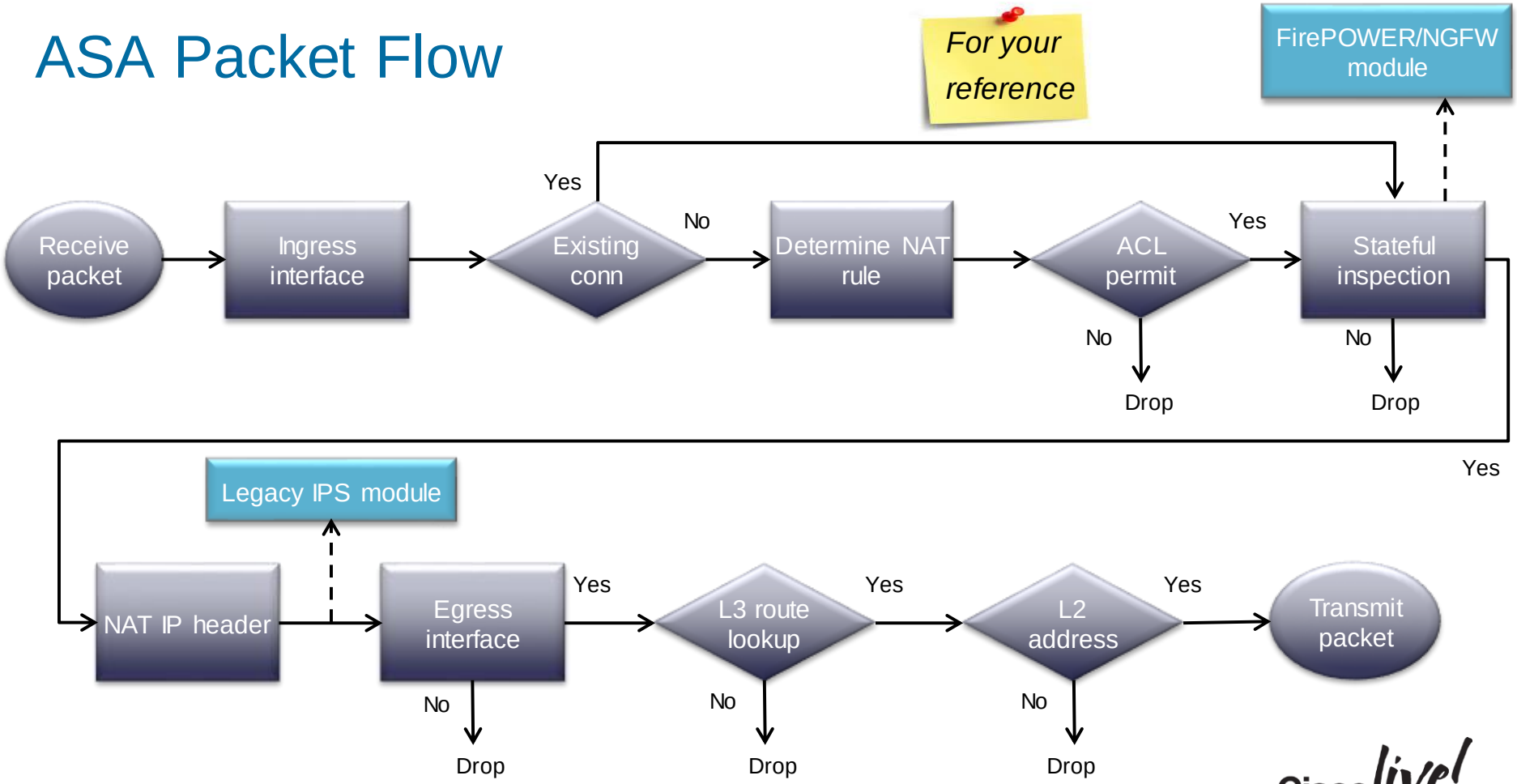
Example Flow

- 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**

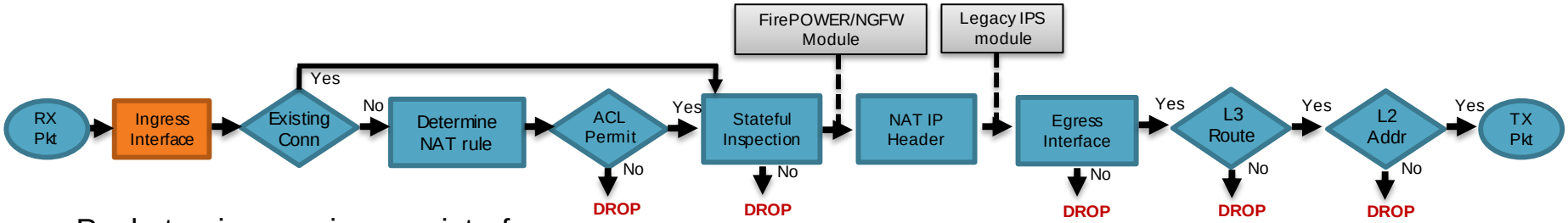


With the Flow defined, examination of configuration issues boils down to just the two Interfaces: **Inside** and **Outside**

ASA Packet Flow



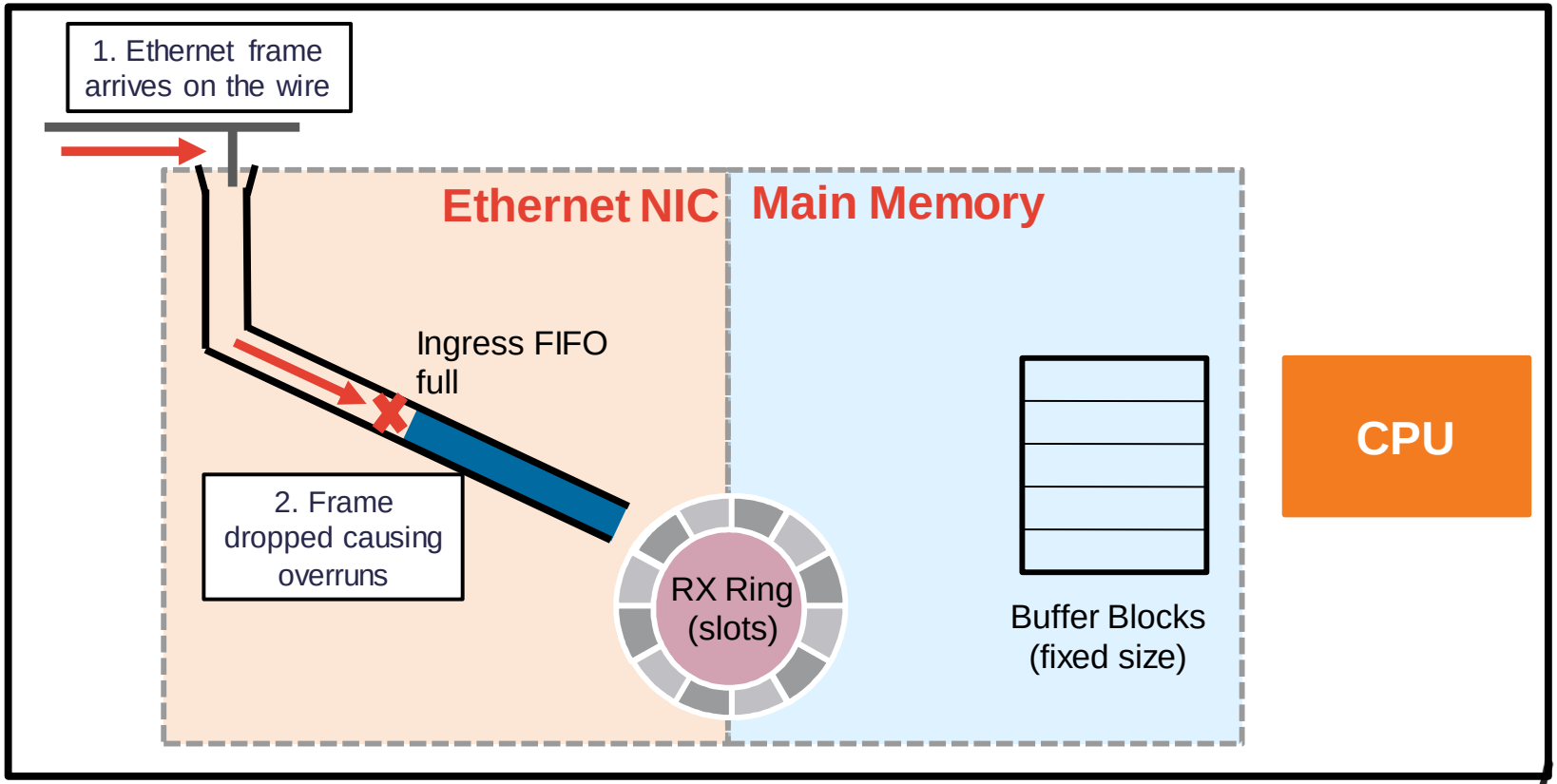
Packet Processing: Ingress Interface



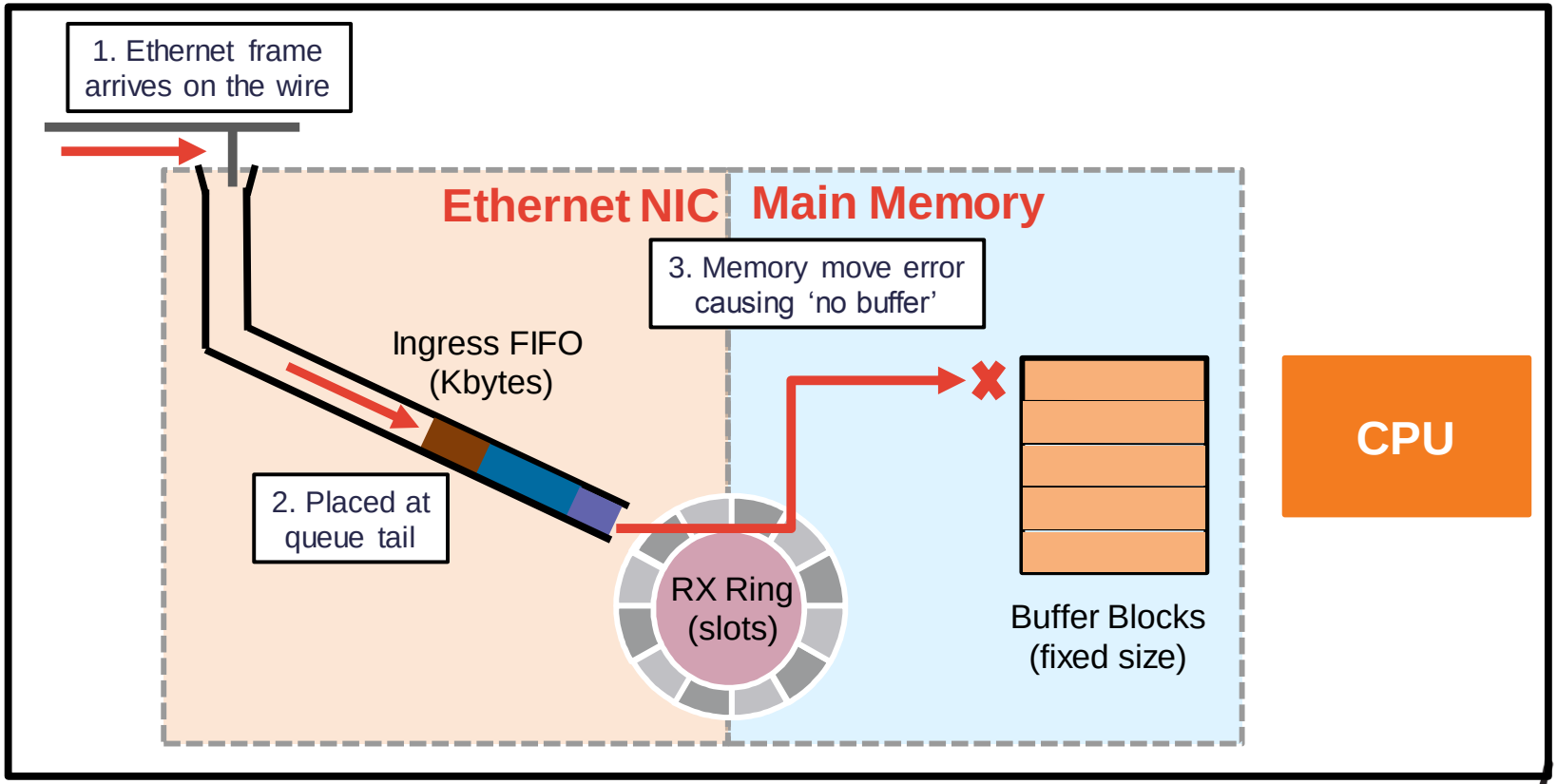
- Packet arrives on ingress interface
- Input counters incremented by NIC and periodically retrieved by CPU
- Software input queue (RX ring) is an indicator of packet load

```
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)
```

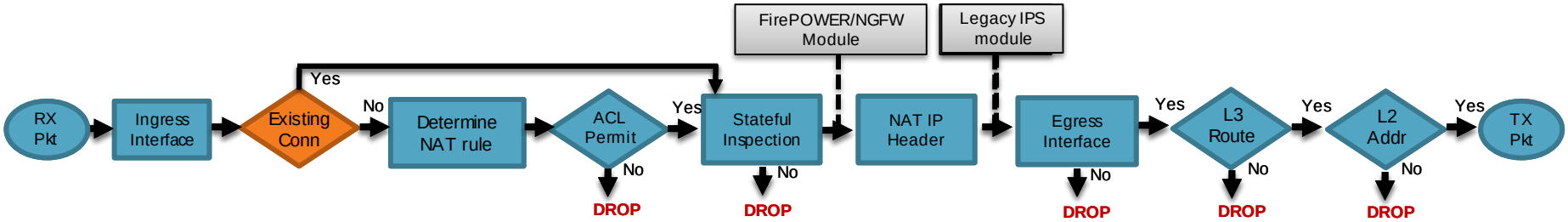
Packet Processing: Ingress Interface



Packet Processing: Ingress Interface



Packet Processing: Locate Connection



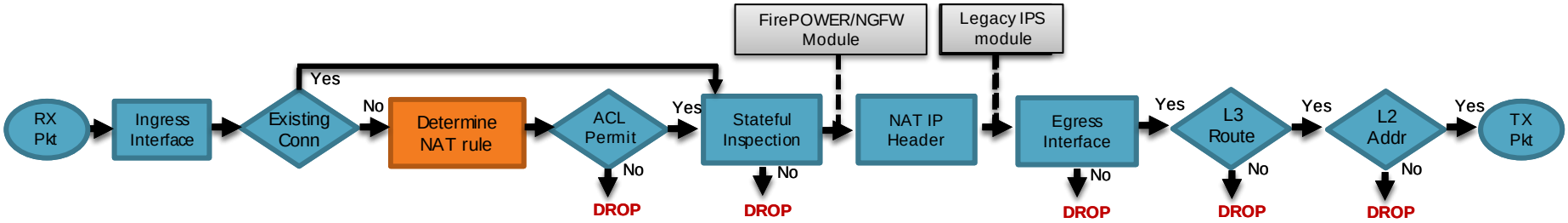
- Check first for existing connection in conn table
- If conn entry exists, bypass ACL check and process in Fastpath

```
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: Determine NAT Rule



- 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

Object-NAT (Auto-NAT)

- Object NAT is the simplest form of NAT, and is defined *within an object*

Host NAT

```
object network obj-WebServer
 host 10.3.19.50
 nat (inside,outside) static 198.51.100.50
```

Network NAT

```
object network Servers
 subnet 10.0.54.0 255.255.255.0
 nat (inside,outside) static 203.0.113.0
```

Dynamic PAT (interface overload)

```
object network InternalUsers
 subnet 192.168.2.0 255.255.255.0
 nat (inside,outside) dynamic interface
```

Twice-NAT

- Twice NAT can specify the source and the destination translation

Network Objects

```
object network 10.10.10.0-net
 subnet 10.10.10.0 255.255.255.0
!
object network 192.168.1.0-net
 subnet 192.168.1.0 255.255.255.0
```

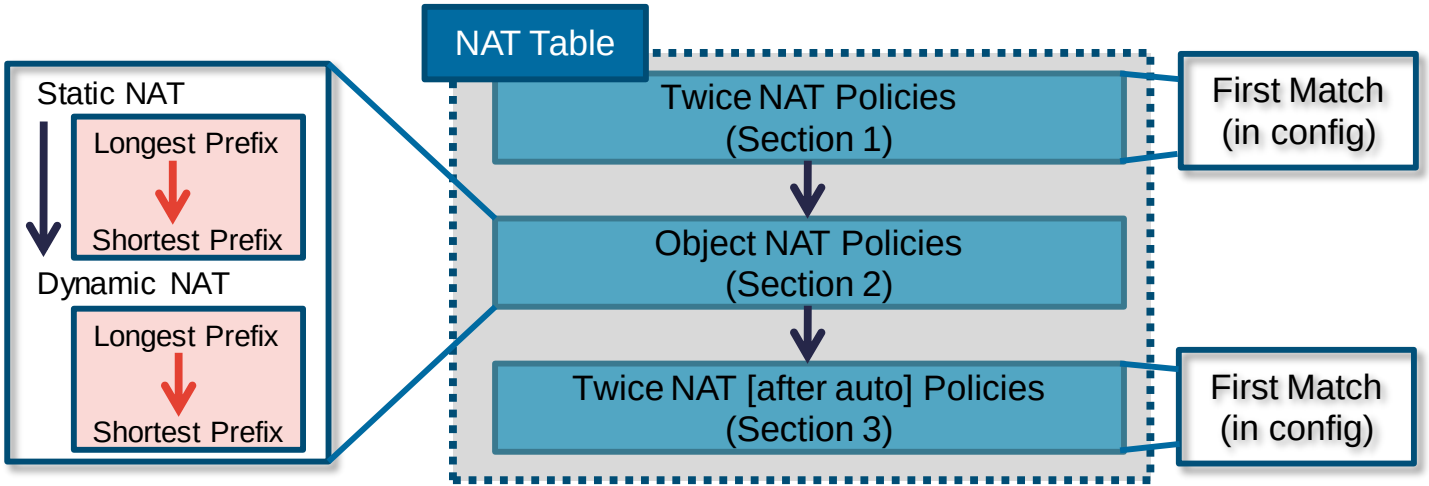
Twice NAT Config

```
nat (inside,outside) source static 10.10.10.0-net 10.10.10.0-net
destination static 192.168.1.0-net 192.168.1.0-net
```

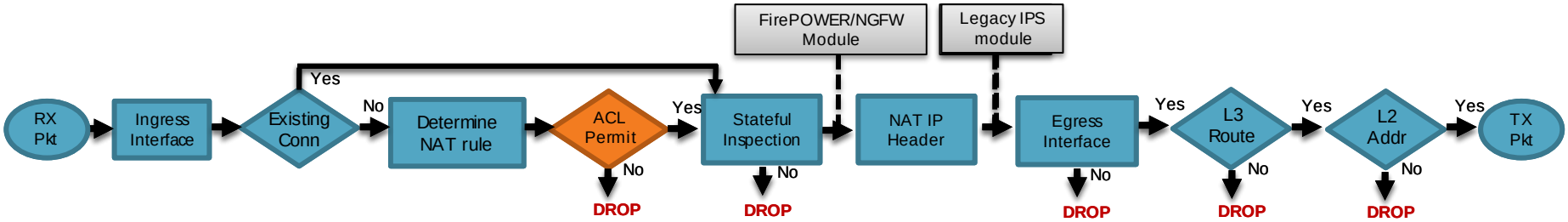
NAT Order of Operation version 8.3+

For your reference

- The ASA configuration is built into the NAT table
- The NAT Table is based on *First Match* (top to bottom)
- The show nat command will display the NAT table in order



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

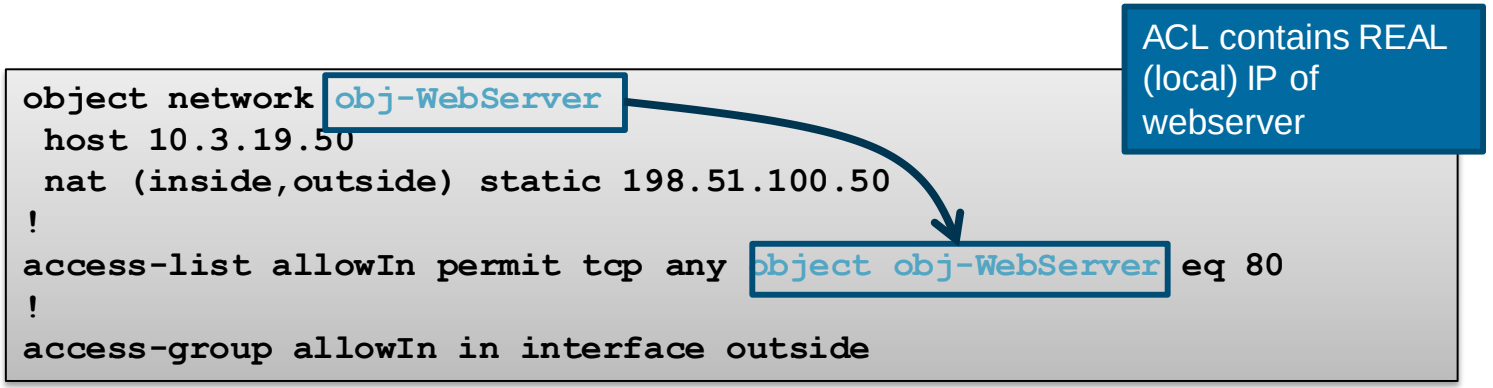
```
asa# show access-list inside
access-list inside line 10 permit ip 10.1.1.0 255.255.255.0 any (hitcnt=1)
```

- 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"
```

Real-IP Used by ACL

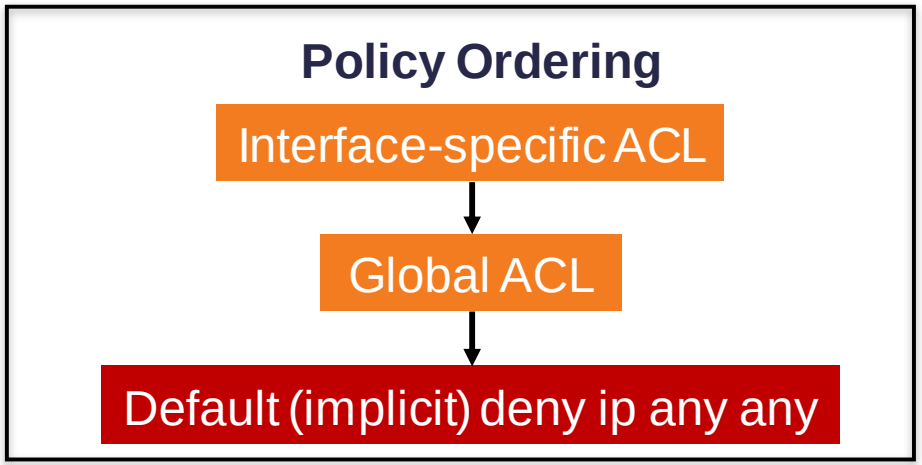
- A reminder that with 8.3+ *Real-IPs* are used in ACLs



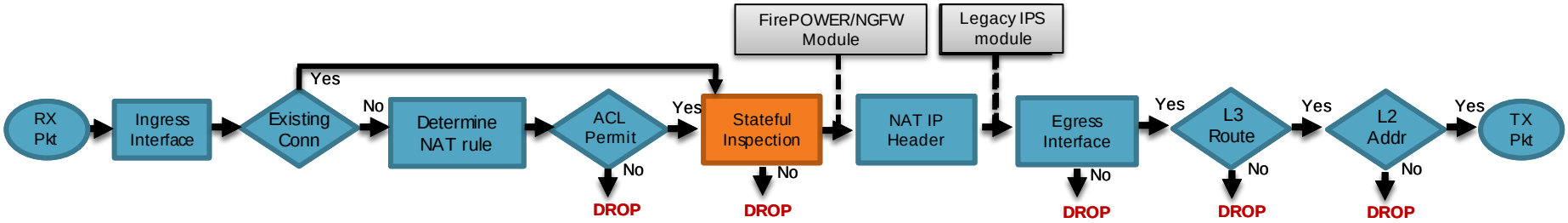
Global ACLs

- 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
```



Packet Processing: Stateful Inspection

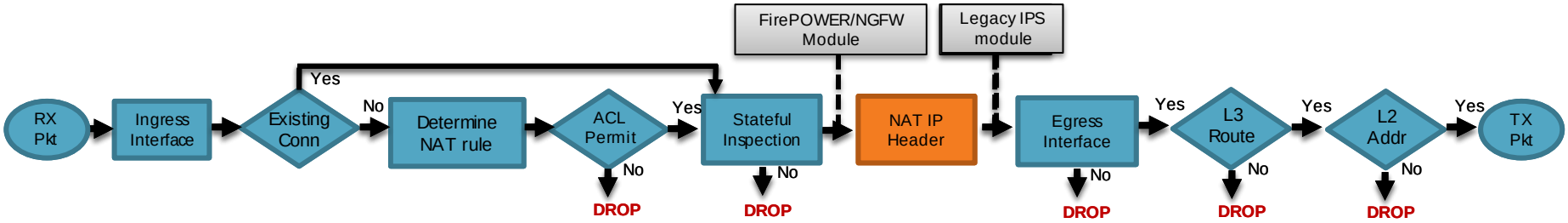


- Stateful inspection ensures very little protocol compliance
- (Optional) Customisable application inspection up to Layer 7 (FTP, SIP, and so on)
  - Rewrite embedded IP addresses, open up ACL pinholes for secondary connections
  - Additional security checks are applied to the application payload

```
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
```

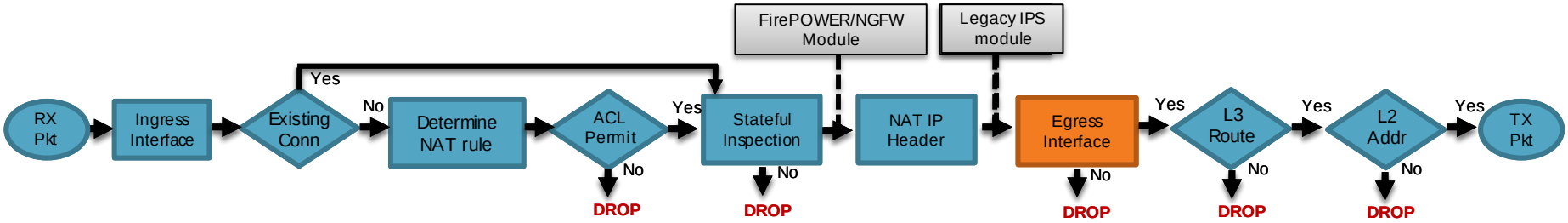
- Packets forwarded to FirePOWER/NGFW services module at this stage.
  - NAT information carried in proprietary header.

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 legacy IPS module
  - Real (pre-NAT) IP address information is supplied as meta data

Packet Processing: Egress Interface



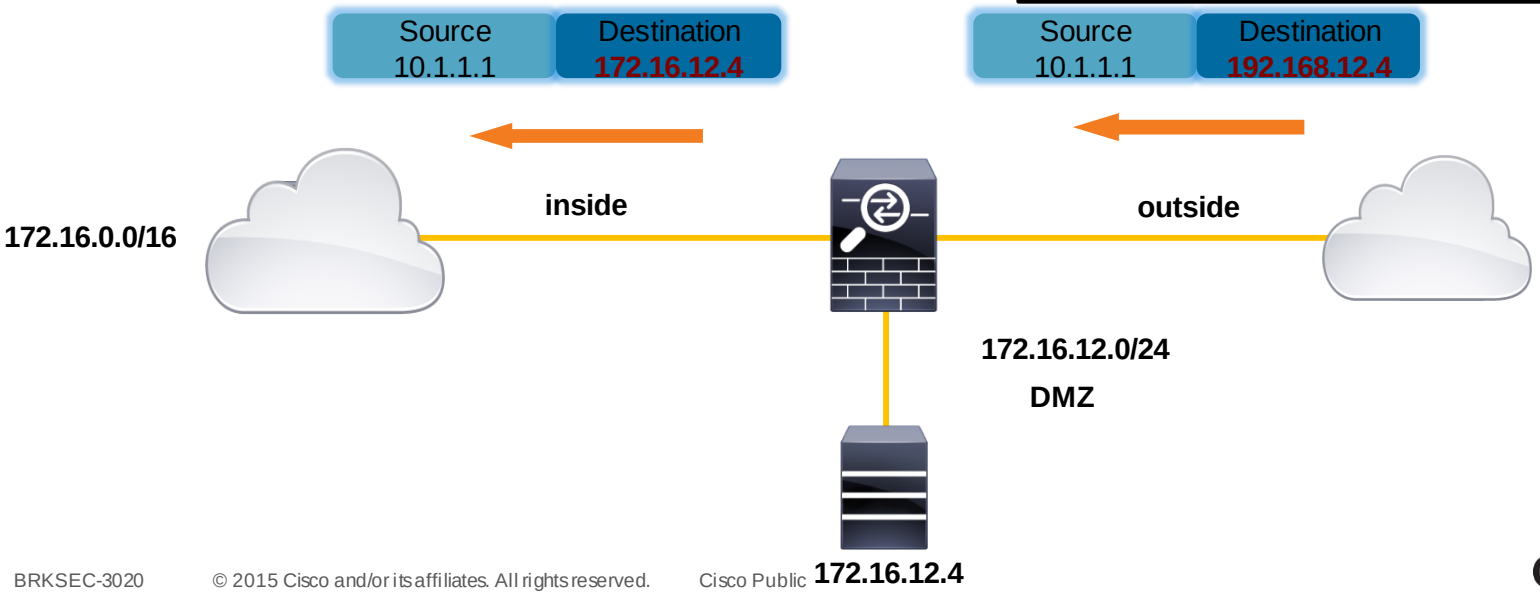
- Packet is virtually forwarded to egress interface (not forwarded to the Ethernet NIC yet)
- Egress interface is determined first by existing conn entry or translation rules, only THEN the routing table
- If NAT does not divert to the egress interface, the global routing table is consulted to determine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

NAT Traffic Diversion

Where would this packet go?

```
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
```

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 Traffic Diversion

- Network Object and Twice NAT override routing table on inbound
  - Network Object NAT **diverts** packets to real interface in only one direction.

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

Actual translation, 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 bidirectionally.

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**

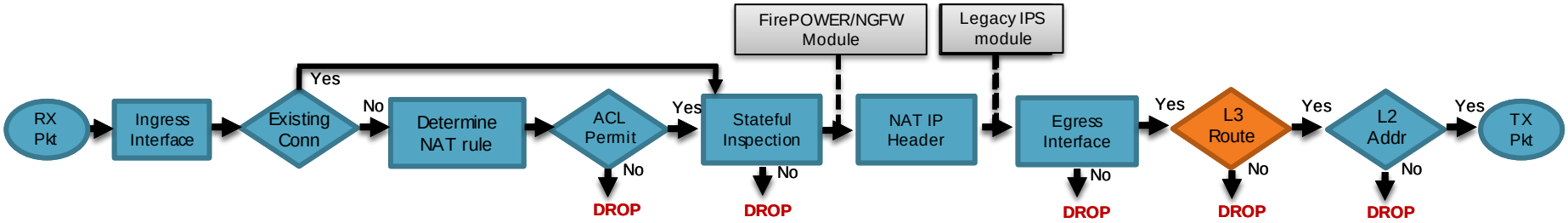
```
nat (inside,outside) source static 192_168_1_0 192_168_1_0 destination static 192_168_2_0
192_168_2_0
```

- 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 for **10.0.0.0** would be diverted to **inside** without '**route-lookup**' – use routing table for egress interface determination

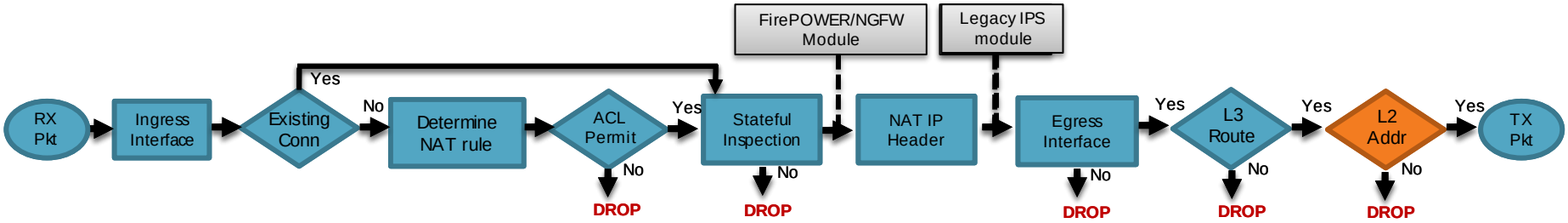
Packet Processing: L3 Route Lookup



- 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
```

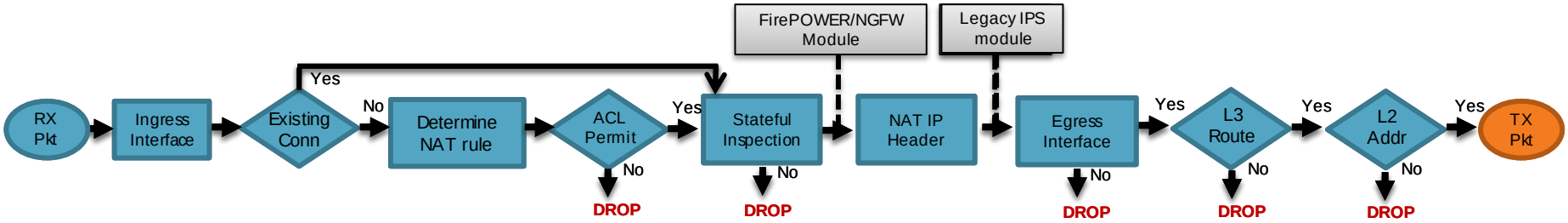
Packet Processing: L2 Address Lookup



- 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



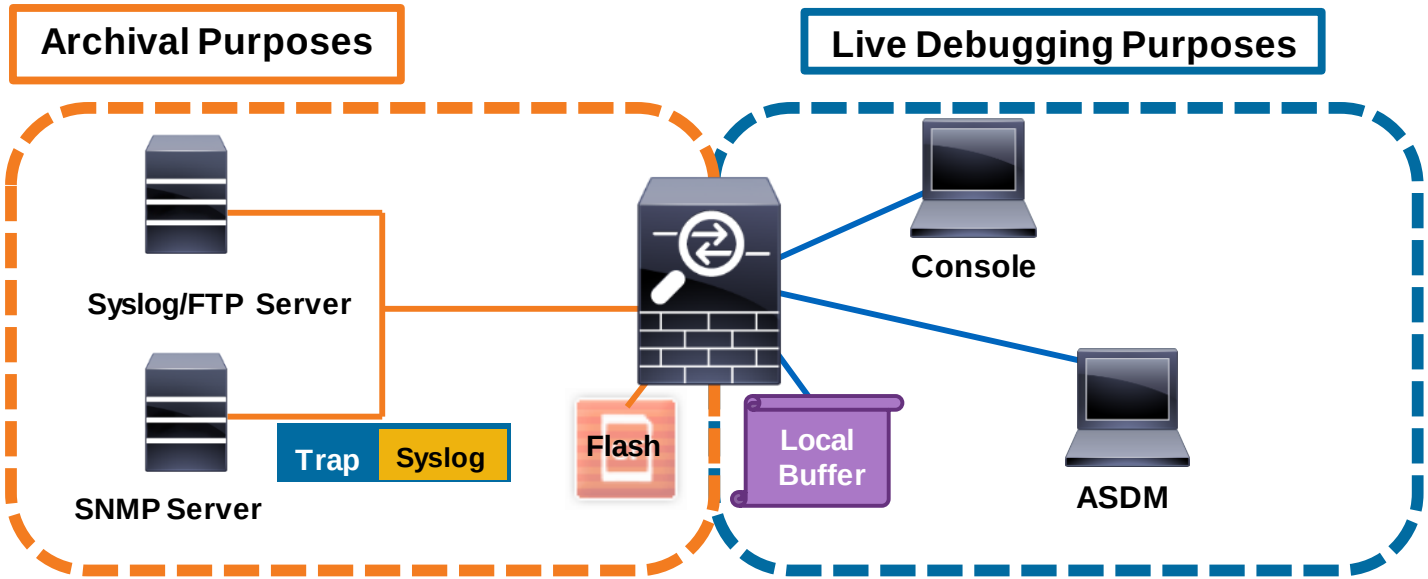
- Packet is transmitted on wire
- Interface counters will increment on interface
- **Underrun** counter indicates drops due to egress interface oversubscription
  - TX ring is full

```
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)
```

Troubleshooting Tools

Uses of Syslogs

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



ASA Syslog Level vs. Number of Messages

Log Level	Description	Number of Messages (SUM)							
		Ver. 7.0	Ver. 7.2	Ver. 8.0	Ver. 8.1	Ver. 8.2	Ver. 8.3	Ver. 8.4	Ver. 9.1
0	Emergencies	0	0	0	0	0	0	0	0
1	Alerts	62 (62)	77 (77)	78 (78)	87 (87)	87 (87)	95 (95)	109 (109)	117 (117)
2	Critical	29 (91)	35 (112)	49 (127)	50 (137)	56 (143)	57 (152)	63 (172)	72 (189)
3	Errors	274 (365)	334 (446)	361 (488)	363 (500)	384 (527)	408 (560)	448 (620)	521 (710)
4	Warnings	179 (544)	267 (713)	280 (768)	281 (781)	315 (842)	324 (884)	357 (997)	420 (1130)
5	Notifications	161 (705)	206 (919)	216 (984)	218 (999)	237 (1079)	246 (1130)	265 (1242)	285 (1415)
6	Informational	234 (939)	302 (1221)	335 (1319)	337 (1336)	368 (1447)	377 (1507)	395 (1637)	430 (1845)
7	Debugging	217 (1156)	258 (1479)	266 (1585)	267 (1603)	269 (1716)	269 (1776)	276 (1913)	295 (2140)

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

Cisco live!

Logging – Common Issues

- logging flash-bufferwrap should only be used when logging to buffer at Level 1
- logging history should only be used when you really have an SNMP server that you want to receive all syslogs
- logging console should only be enabled while actively troubleshooting on the console
- logging standby should only be used if you want to receive double the syslogs
- logging permit-hostdown should always be used with TCP syslogging

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>
```

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.

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

Cisco live!

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
```

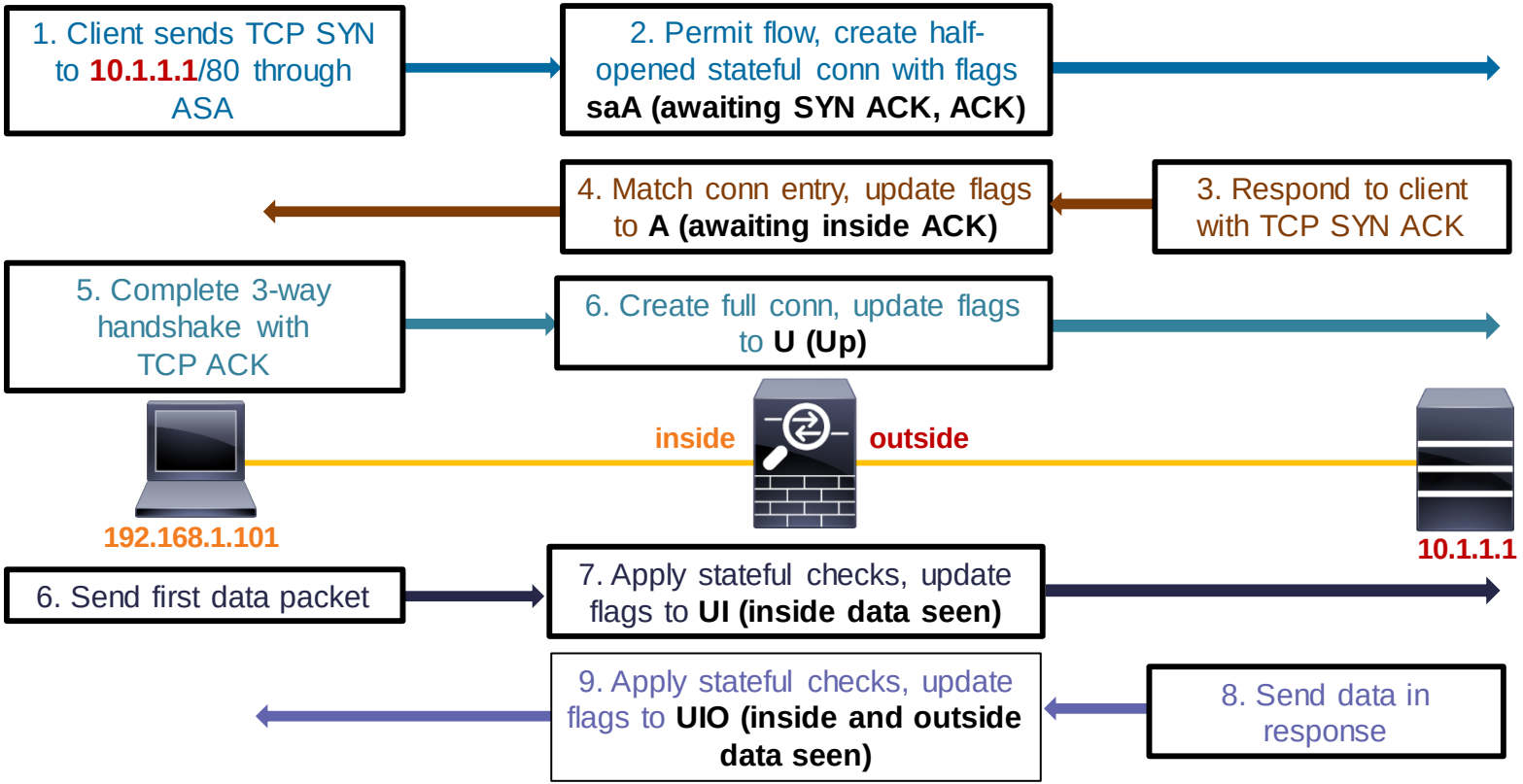
Narrow down the output with **show conn address <ip>**

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

Conn flags indicate current state

detail option adds uptime and timeout information

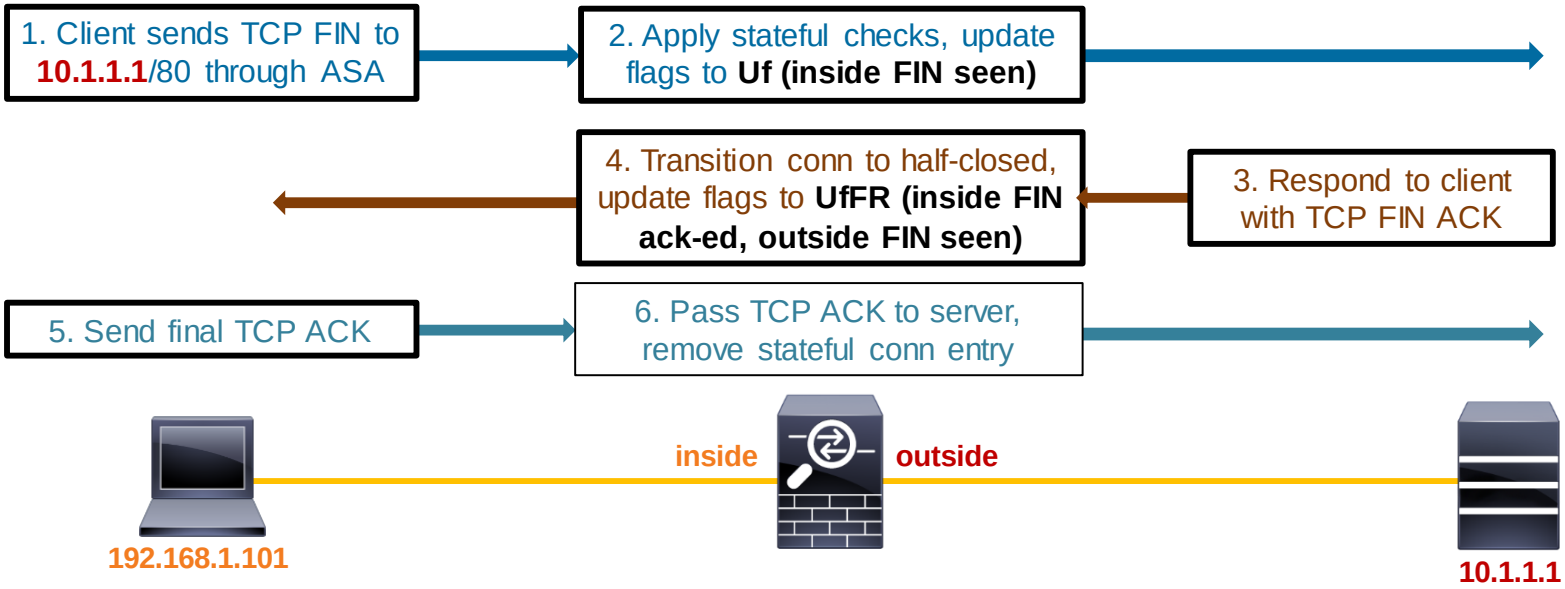
Example: Connection Establishment




```
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
```

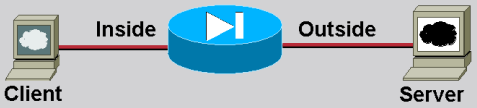


Connection Flags

For your reference

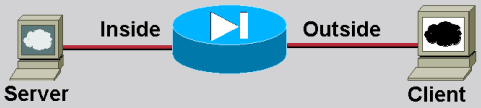
Outbound Connection

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

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

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)
```

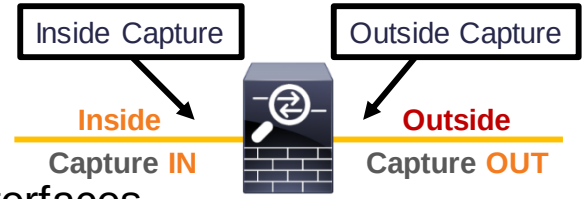
Only display hosts that have more than 50 active TCP connections.

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
   10942
   Bad TCP Checksum (bad-tcp-cksum)                           893

…
```

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

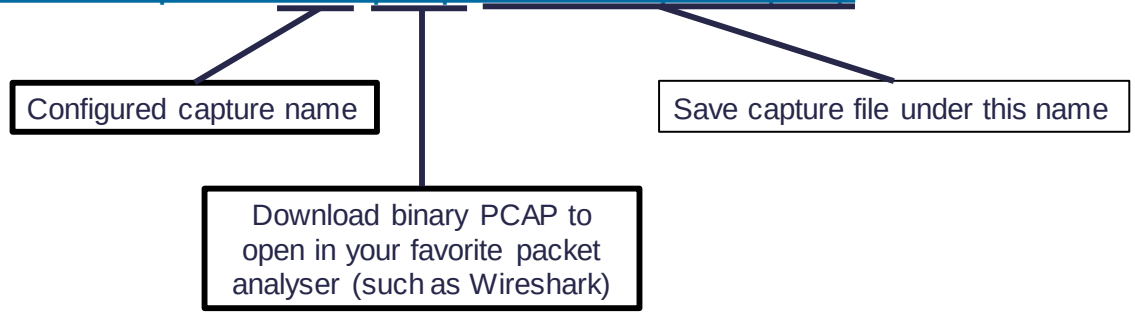
```
asa# capture OUT interface outside match ip any host 172.18.124.1
asa# capture IN interface inside match ip any host 172.18.124.1
asa# show capture IN

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

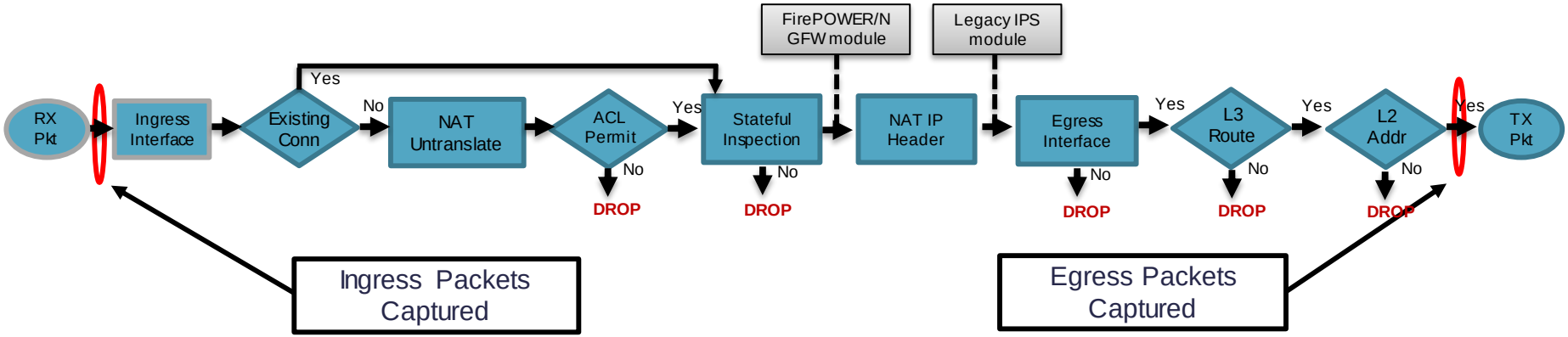
Remember to remove the captures when done with troubleshooting

Packet Capture

- Capture buffer maintained in RAM (512KB by default, 32 MB max)
  - 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
  - https://x.x.x.x/admin/capture/OUT/pcap/outsidecapture.pcap



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** most 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 Dropped Packets

- Capture all frames dropped in the ASP

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

- Capture all frames with a specific drop reason

```
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
```

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

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

Capturing Internal Traffic

- Ability to capture on internal interfaces of the ASA:
  - Only **asa_dataplane** supports filtering using match/access-list.

Used to redirect traffic from ASA to software/hardware module

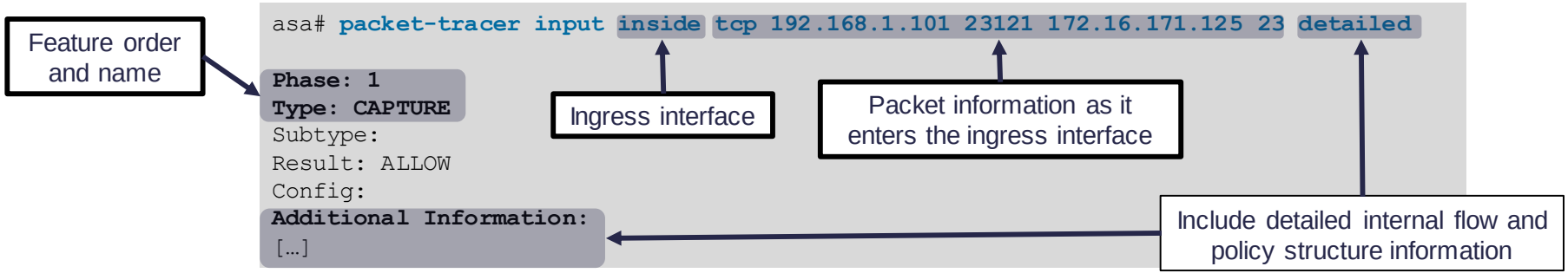
Only for 5500-x – Used by ASA to send and receive management traffic for software module

Used for control communication between ASA and software/hardware module

```
ciscoasa# capture cap interface ?

  asa_dataplane   Capture packets on dataplane interface
  asa_mgmt_plane  Capture packets on managementplane interface
  cplane          Capture packets on controlplane interface
```

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



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
…….
```

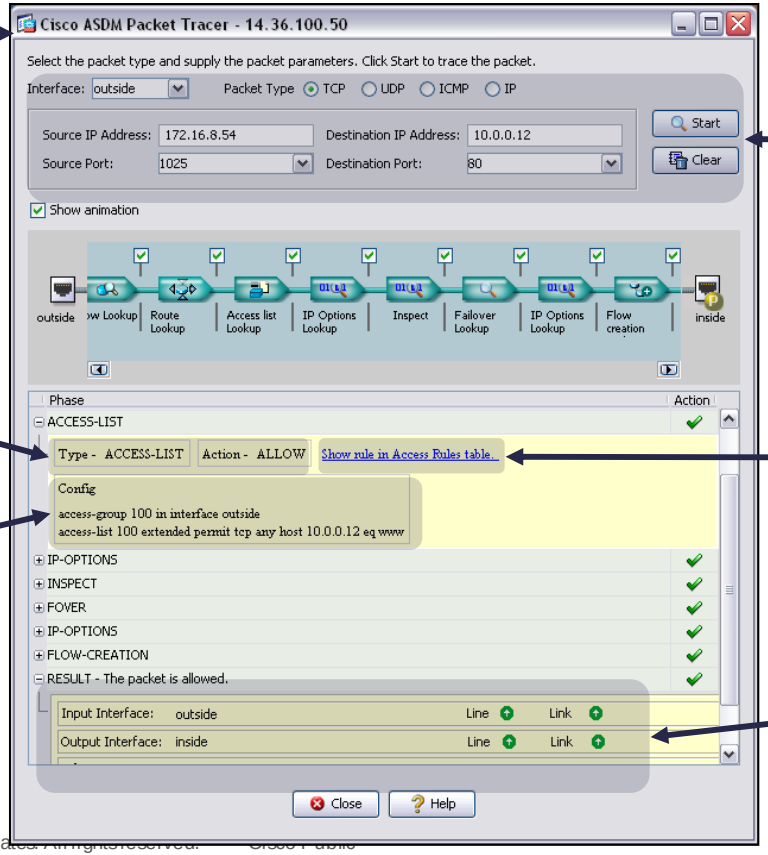
Sample Packet Tracer Output

```
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

For your reference

Launch from **Tools > Packet Tracer**



Define simulated packet

Feature type and resulting action

Direct link to edit policy

Associated configuration

Final outcome (allowed or dropped) and egress interface information

Cisco live!

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

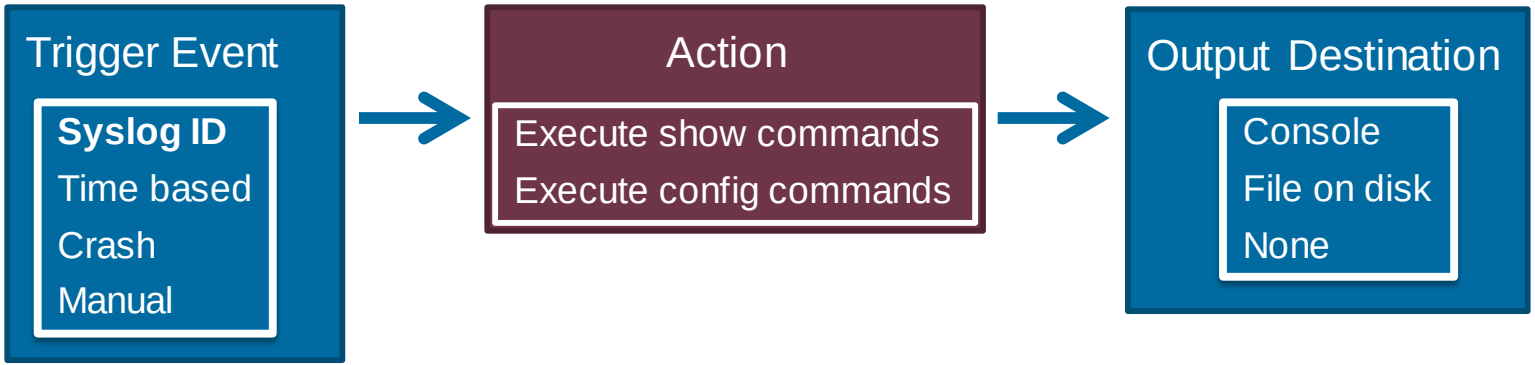
```
asa# show capture inside
  68 packets captured
  1: 15:22:47.581116 10.1.1.2.31746 > 198.133.219.25.80: S
  2: 15:22:47.583465 198.133.219.25.80 > 10.1.1.2.31746: S ack
  3: 15:22:47.585052 10.1.1.2.31746 > 198.133.219.25.80: . ack
  4: 15:22:49.223728 10.1.1.2.31746 > 198.133.219.25.80: P ack
 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

Embedded Event Manager

- Troubleshooting tool added in 9.2(1), similar to IOS EEM
- Powerful way to run CLI commands based on ASA events (syslogs) and save the output



Embedded Event Manager

- Time-based events
  - Every midnight back up the ASA configuration to your tftp server
  - Every 3 hours gather the output of 'show memory detail' and save it to the flash
- Syslog based events
  - If the available 1550 byte blocks become depleted, gather 'show blocks pool 1550 dump' and save to the disk
  - If the AAA server is marked down: 'ping tcp' to the server on port 49, 'show aaa-server' to gather statistics, save to a file on disk, use SCH to email the file contents
- Manual events
  - Gather the output of 10 different commands and save to a file

Embedded Event Manager

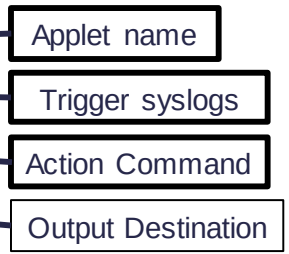
- Goal: Backup the configuration when a user logs in, and again when they log off of a SSH session
  - Determine the syslogs that should trigger the event

```
%ASA-6-605005: Login permitted from 14.36.103.220/54785 to 36net:14.36.103.88/ssh for user "cisco"
```

```
%ASA-5-611103: User logged out: Uname: cisco
```

  - Configure the event applet

```
event manager applet loginConfigBackup
 event syslog id 605005
 event syslog id 611103
 action 1 cli command "show running-config"
 output file rotate 50
!
```



  - Files written to disk when a user logs in and then out

```
261    -rwx  161286       16:46:27 May 05 2014  eem-loginConfigBackup-0.log
260    -rwx  161331       16:46:14 May 05 2014  eem-loginConfigBackup-1.log
259    -rwx  161277       16:46:07 May 05 2014  eem-loginConfigBackup-2.log
```

Case Study:
ASA Frequent Failover

Problem Description

- ASAs in failover are failing over frequently at certain times of the day.
- Reason for failover is interface check failure.


```
asa/pri# show failover history
==========================================================
From State       To State         Reason
==========================================================
23:37:28 UTC Oct 13 2014
Active           Failed           Interface check
23:37:29 UTC Oct 13 2014
Failed           Standby Ready    Interface check
23:37:45 UTC Oct 13 2014
Standby Ready    Just Active      Other unit wants me Active
```

```
asa/sec# show failover history
==========================================================
From State       To State         Reason
==========================================================
23:37:28 UTC Oct 13 2014
Standby Ready    Just Active      Other unit wants me Active
23:37:45 UTC Oct 13 2014
Active           Failed           Interface check
23:37:45 UTC Oct 13 2014
Failed           Standby Ready    Interface check
```

ASAs are failing over few times a minute but recover themselves within a second

Verifying Syslogs

- Quick way to check which interface is failing at the time of failover

```
Oct 13 2014 23:37:59 asa : %ASA-1-105005: (Primary) Lost Failover communications with mate on
interface inside
Oct 13 2014 23:37:59 asa : %ASA-1-105008: (Primary) Testing Interface inside
Oct 13 2014 23:38:01 asa : %ASA-1-105009: (Primary) Testing on interface inside Failed
Oct 13 2014 23:38:01 asa : %ASA-1-104002: (Primary) Switching to STNDBY - Interface check
Oct 13 2014 23:38:02 asa : %ASA-1-104004: (Primary) Switching to OK.
Oct 13 2014 23:50:25 asa : %ASA-1-104001: (Primary) Switching to ACTIVE - Other unit wants me
Active. Secondary unit switch reason: Interface check.
```

Interface Status and Stats

- **show interface** to verify interface status and statistics

```
Interface GigabitEthernet0/0 "inside", is up, line protocol is up
  Hardware is i82598af rev01, BW 10000 Mbps, DLY 10 usec
        (Full-duplex), (1000 Mbps)
        Input flow control is unsupported, output flow control is off
        MAC address 001b.2158.2fdb, MTU not set
        IP address 10.1.1.1
        2624988886889 packets input, 2056041800082596 bytes, 0 no buffer
        Received 11320 broadcasts, 0 runs, 0 giants
        67219561 input errors, 0 CRC, 0 frame, 67219561 overrun, 0 ignored, 0 abort
        0 pause input, 0 resume input
        7098 L2 decode drops
        2003400244878 packets output, 844075019064354 bytes, 5 underruns
        0 pause output, 0 resume output
        0 output errors, 0 collisions, 2 interface resets
        0 late collisions, 0 deferred
        0 output decode drops
        0 input reset drops, 0 output reset drops
```

Large number of overruns could be the cause of failover. Needs further investigation

Periodic Output Collection

- **EEM Applet** to run commands periodically and store them in a file.

```
asa# sh run event manager
event manager applet COLLECT-OUTPUTS
 event timer watchdog time 60
 action 0 cli command "show failover history"
 action 1 cli command "show interface detail"
 action 2 cli command "show perfmon"
 action 3 cli command "show process cpu-usage
sorted non-zero"
 output file append disk0:/primary-output.txt
```

CLI commands to execute every 60 seconds

- **EEM Applet** to copy file to remote server on detecting failover.

```
event manager applet EMAIL-OUTPUT
 event syslog id 105009
 action 0 cli command "copy /noconfirm
flash:/primary-output.txt
tftp://10.76.76.160/primary-output.txt"
 output none
```

Detect failover syslog and copy files to remote TFTP server

Analysing Output File

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

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

```
asa# show perfmon

PERFMON STATS:                          Current       Average
Xlates                                    0/s           0/s
Connections                             601/s           0/s
TCP Conns                               326/s           0/s
UDP Conns                               218/s           0/s
URL Access                                0/s           0/s
URL Server Req                            0/s           0/s
TCP Fixup                                 4/s           0/s
TCP Intercept Established Conns           0/s           0/s
TCP Intercept Attempts                    0/s           0/s
TCP Embryonic Conns Timeout            6263/s           0/s
HTTP Fixup                                0/s           0/s
FTP Fixup                                20/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         97.85%
```

Checking Incomplete TCP Connection Source

- Add **show conn** to EEM Applet to see who is creating the incomplete connections

```
asa# show conn state tcp_embryonic
54764 in use, 54764 most used
TCP dmz 172.16.101.118:443 inside 10.1.1.50:41326, idle 0:00:23, bytes 0, flags SaA
TCP dmz 172.16.36.109:80 inside 10.1.1.50:51436, idle 0:00:13, bytes 0, flags SaA
TCP dmz 172.16.110.202:80 inside 10.1.1.50:15689, idle 0:00:25, bytes 0, flags SaA
TCP dmz 172.16.2.204:443 inside 10.1.1.50:31567, idle 0:00:29, bytes 0, flags SaA
TCP dmz 172.16.106.205:443 inside 10.1.1.50:60243, idle 0:00:02, bytes 0, flags SaA
TCP dmz 172.16.223.63:443 inside 10.1.1.50:23789, idle 0:00:03, bytes 0, flags SaA
TCP dmz 172.16.213.239:80 inside 10.1.1.50:16753, idle 0:00:04, bytes 0, flags SaA
TCP dmz 172.16.75.192:80 inside 10.1.1.50:41213, idle 0:00:06, bytes 0, flags SaA
```

Only display incomplete connections

All connections are from the same inside IP address scanning TCP ports 80 and 443 on the entire DMZ subnet

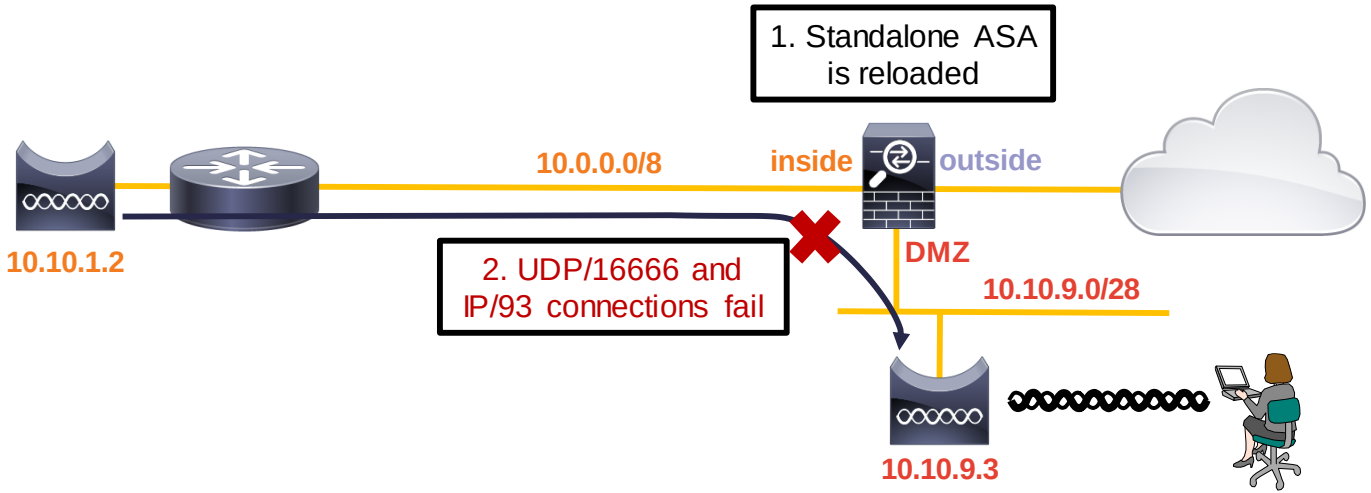
Solution

- An infected host on the local subnet was running port scans at random times of the day causing overruns

Case Study:
UDP Connections Fail After 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



Reviewing Packet Captures and logs

- 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
```

```
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
```

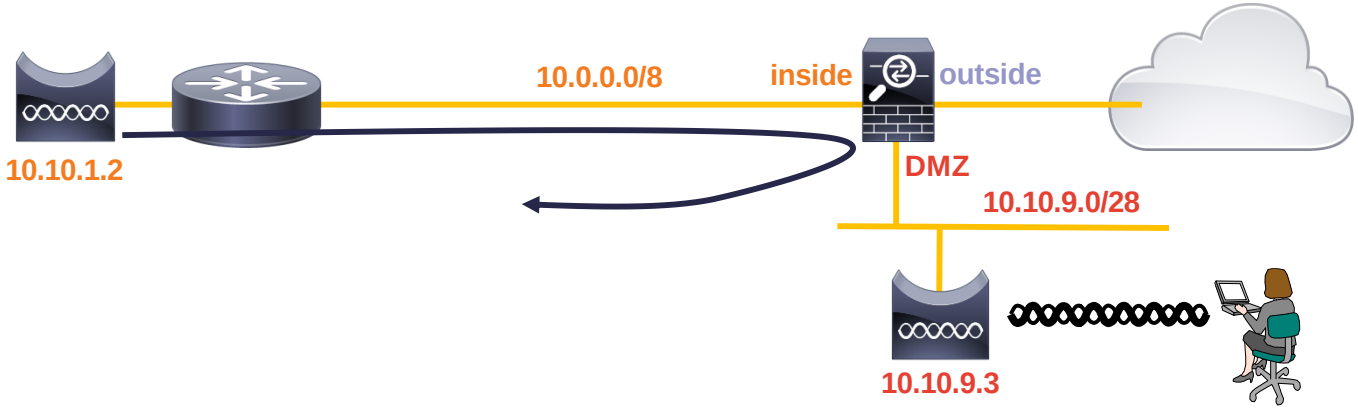
Egress interface capture shows no matching packets

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
```

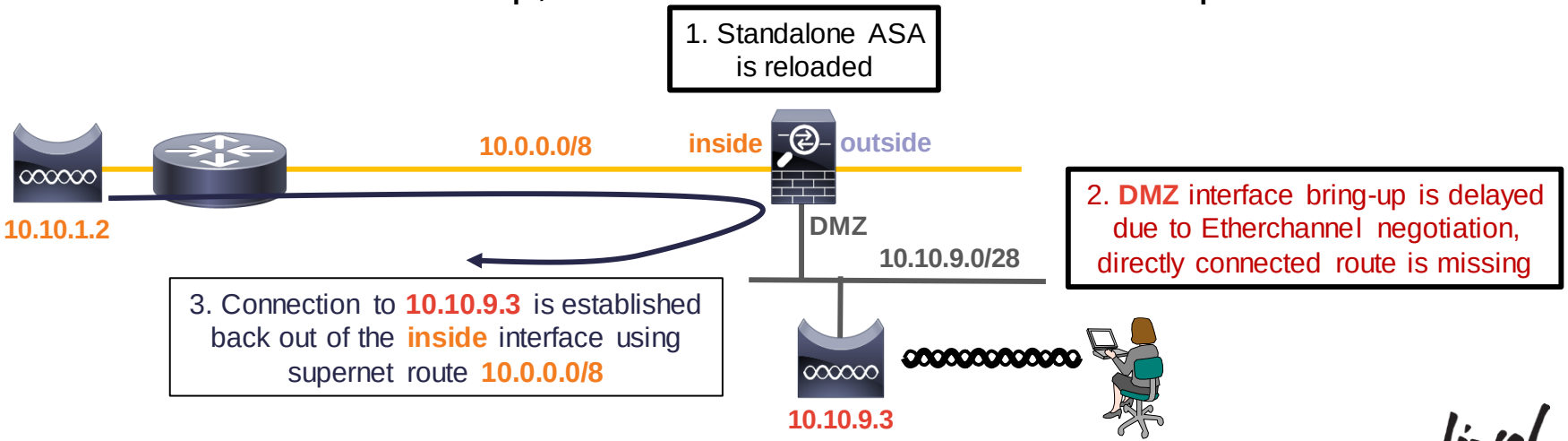
Cisco live!

Root Cause

- When conn entry was created, route lookup for 10.10.9.3 resolved to inside

```
asa# show conn address 10.10.1.2

126 in use, 12654 most used
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
```

- If DMZ interface was not up, the route to 10.10.9.0/28 was not present

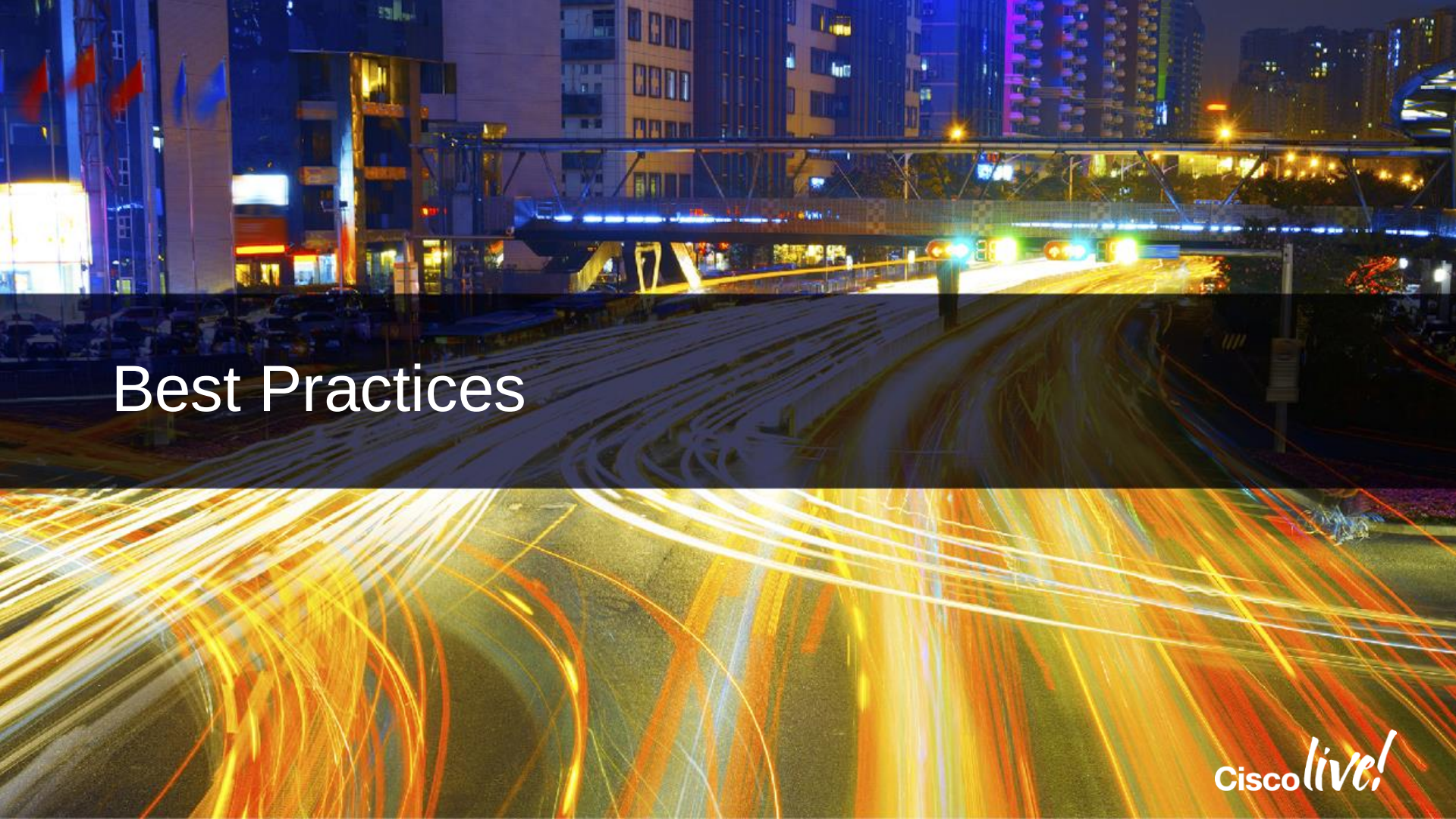


Floating Connection Timeout

- The “bad” connection never times out since the UDP traffic is constantly flowing
  - TCP is stateful, 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



Best Practices

Cisco live!

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

ASA Best Practices

- **Remove** ACL entries that accumulate 0 hit count 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 a variety of tools, including syslogs, **show** commands, Packet Tracer, packet captures


Q & A
Cisco live!

Complete Your Online Session Evaluation

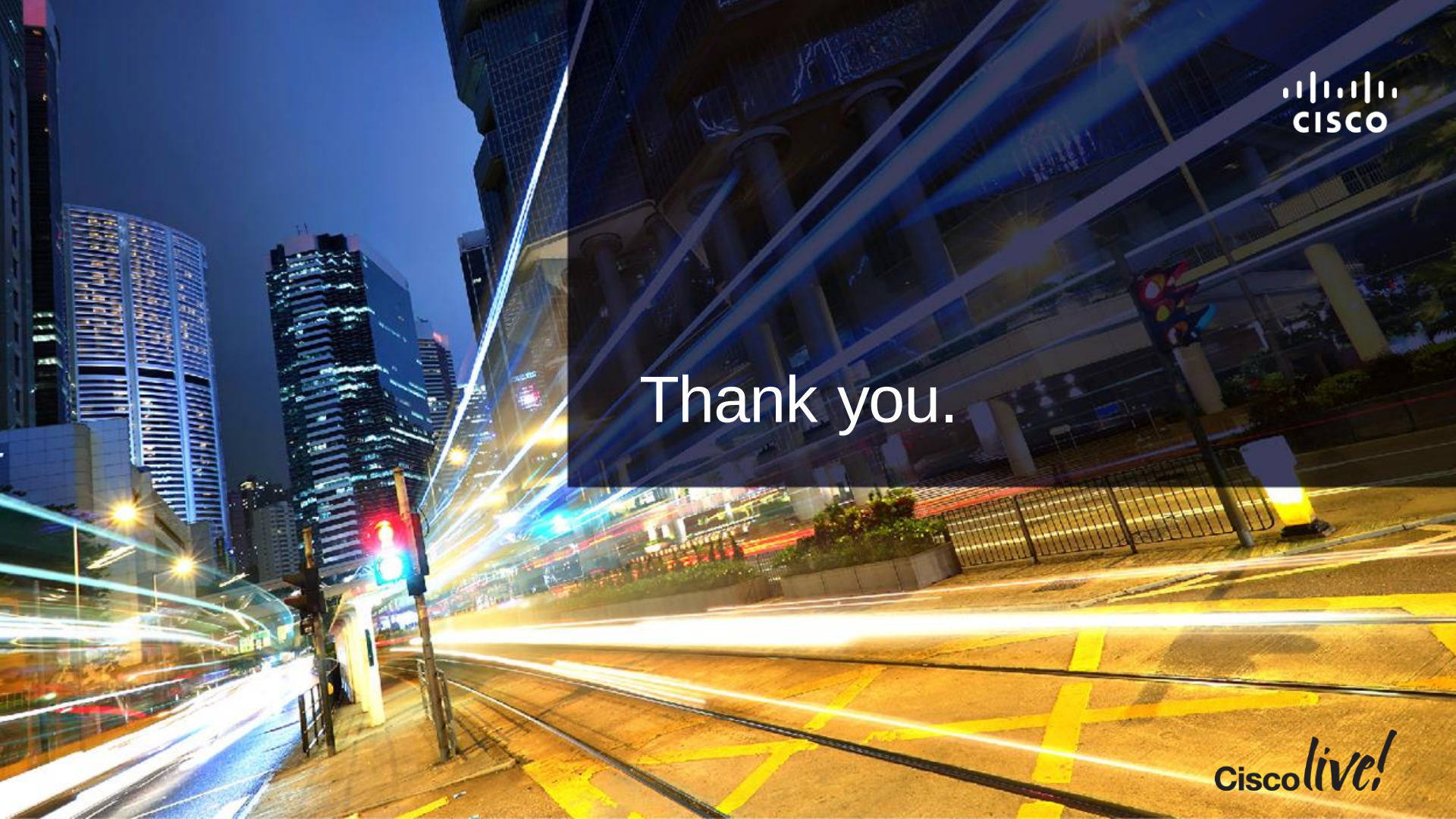
Give us your feedback and receive a Cisco Live 2015 T-Shirt!

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 http://showcase.genie-connect.com/clmelbourne2015
- Visit any Cisco Live Internet Station located throughout the venue

T-Shirts can be collected in the World of Solutions on Friday 20 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


Thank you.
CISCO
Cisco live!


CISCO

Appendix

- Lucky You
- This appendix contains extra information which you may find useful, but I just didn't have enough time to cover in the lecture – or which was covered in previous years.
- Enjoy… :-)

ASA Hardware Architecture

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**

CPU Packet Processing

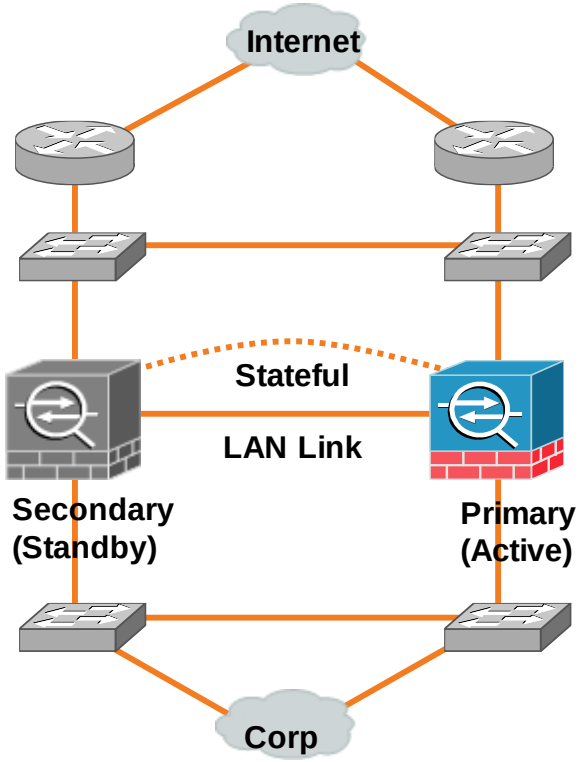
- NIC moves packets from Ethernet to memory
- All packets are processed by the CPU complex in software
- Data Path CPU process checks all inbound packets **sequentially**
  - Stateful checks are applied to every single packet
  - Fastpath, Slowpath, Control Plane
- New connection requests are directed 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

Failover Basics

- Active/Standby vs. Primary/Secondary
- Stateful failover (optional)
- A failover only occurs when either firewall determines the standby firewall is healthier than the active firewall
- Both firewalls swap MAC and IP addresses when a failover occurs
- Level 1 syslogs will give reason of failover



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
```

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

TAC Tips and Tricks - Failover

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

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

- Configure the session prompt to indicate failover unit and state

```
ASA-SM#
ASA-SM(config)# prompt hostname state priority
ASA-SM/act/pri(config)# exit
ASA-SM/act/pri#
```

Active vs. Standby

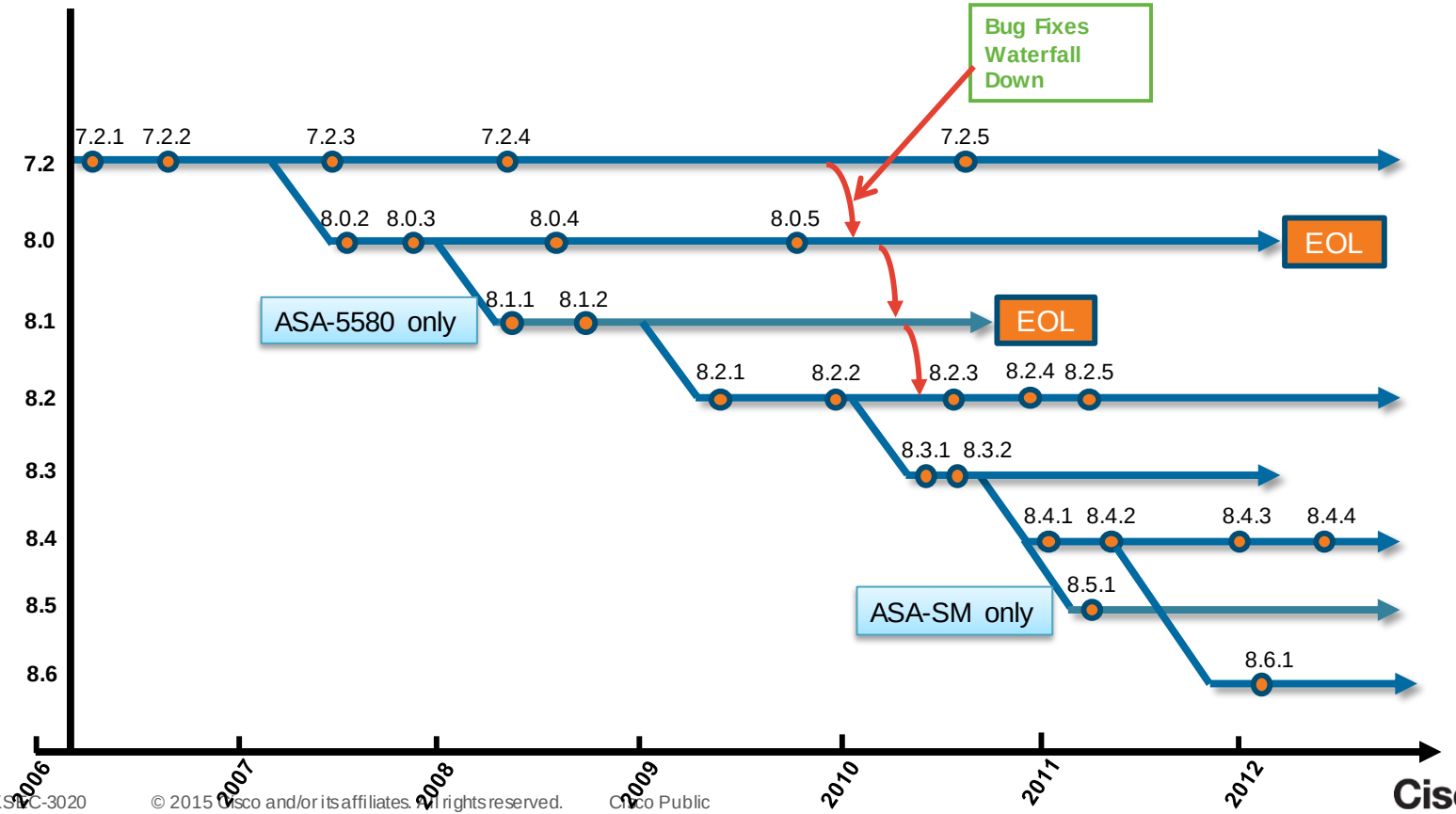
Primary vs. Secondary

Redirecting Debugs to Syslog

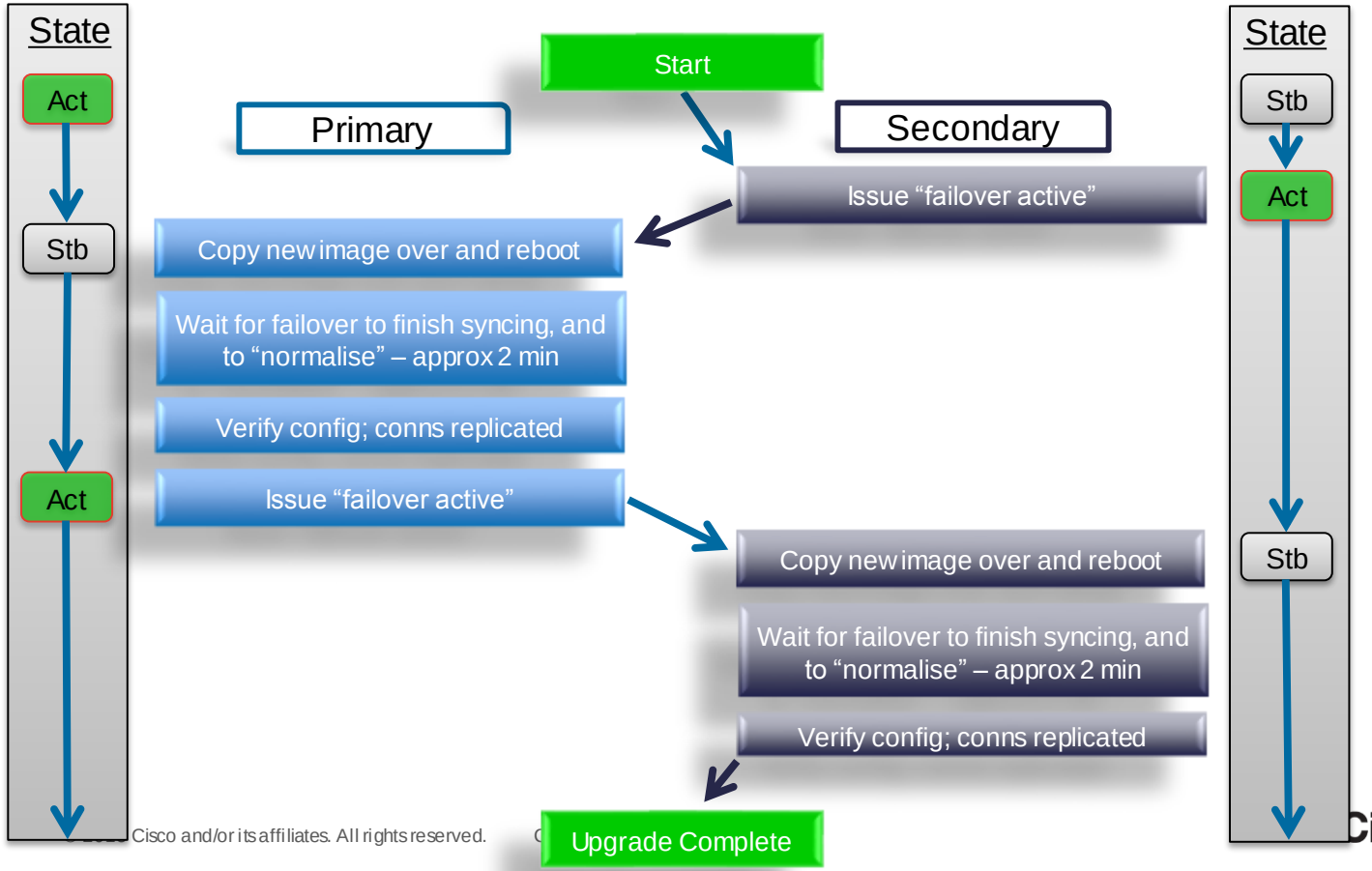
- Problem
  - Log only debug output to syslog
- Solution
  - Create a logging list with only syslog ID 711001
  -
```
ASA(config)# logging list Networkers message 711001
```
  - Enable debug output to syslogs
  -
```
ASA(config)# logging debug-trace
INFO: 'logging debug-trace' is enabled. All debug messages
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



Example: Show Output Filters

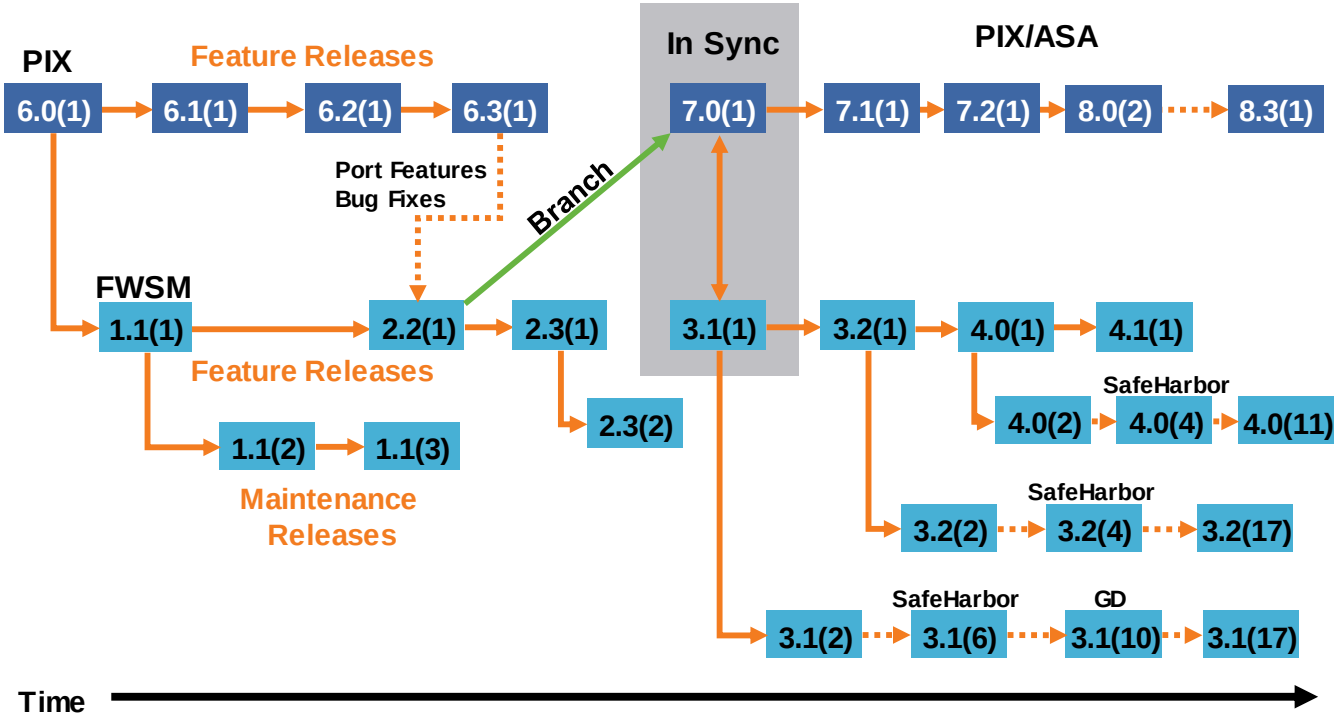
Examples

- 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>
```

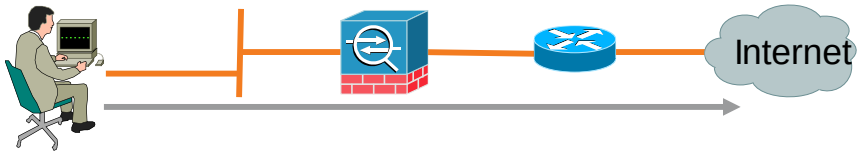
Note: You must Include a Space on Either Side of the Pipe for the Command to Be Accepted; Also, Trailing Spaces Are Counted

Cisco PIX/ASA/FWSM Code Base History



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
```

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

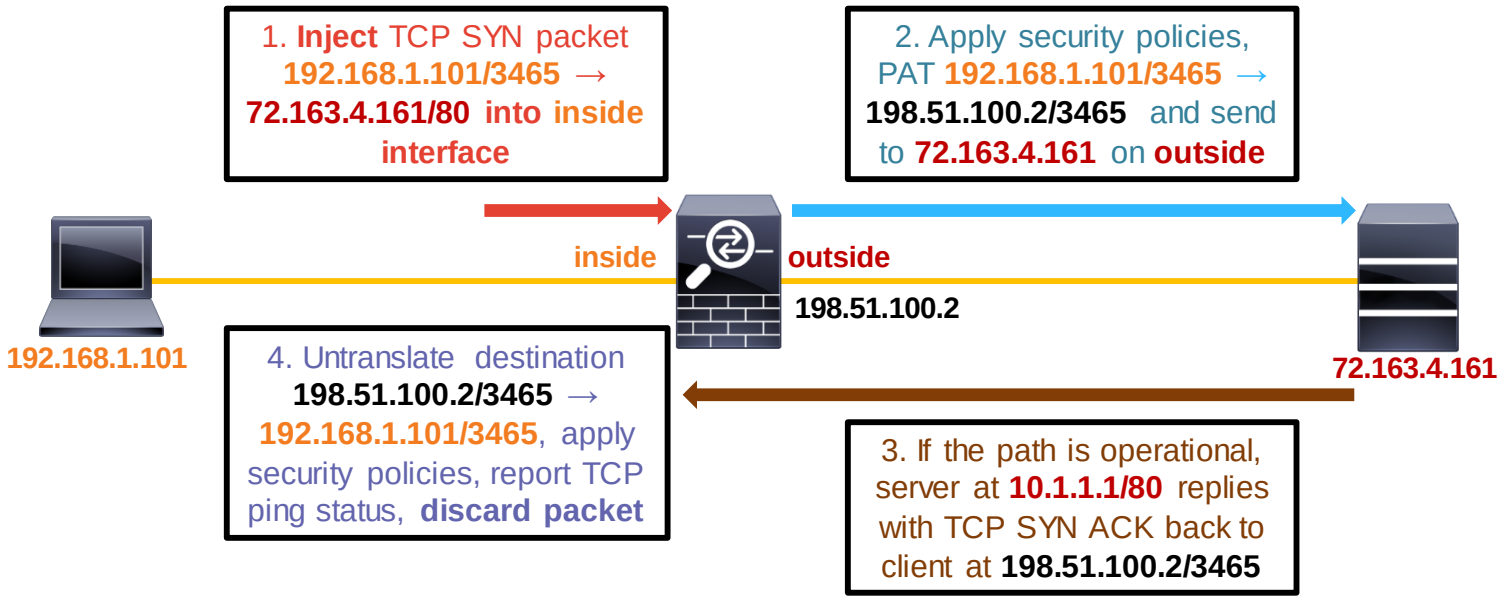
Example: TCP Ping

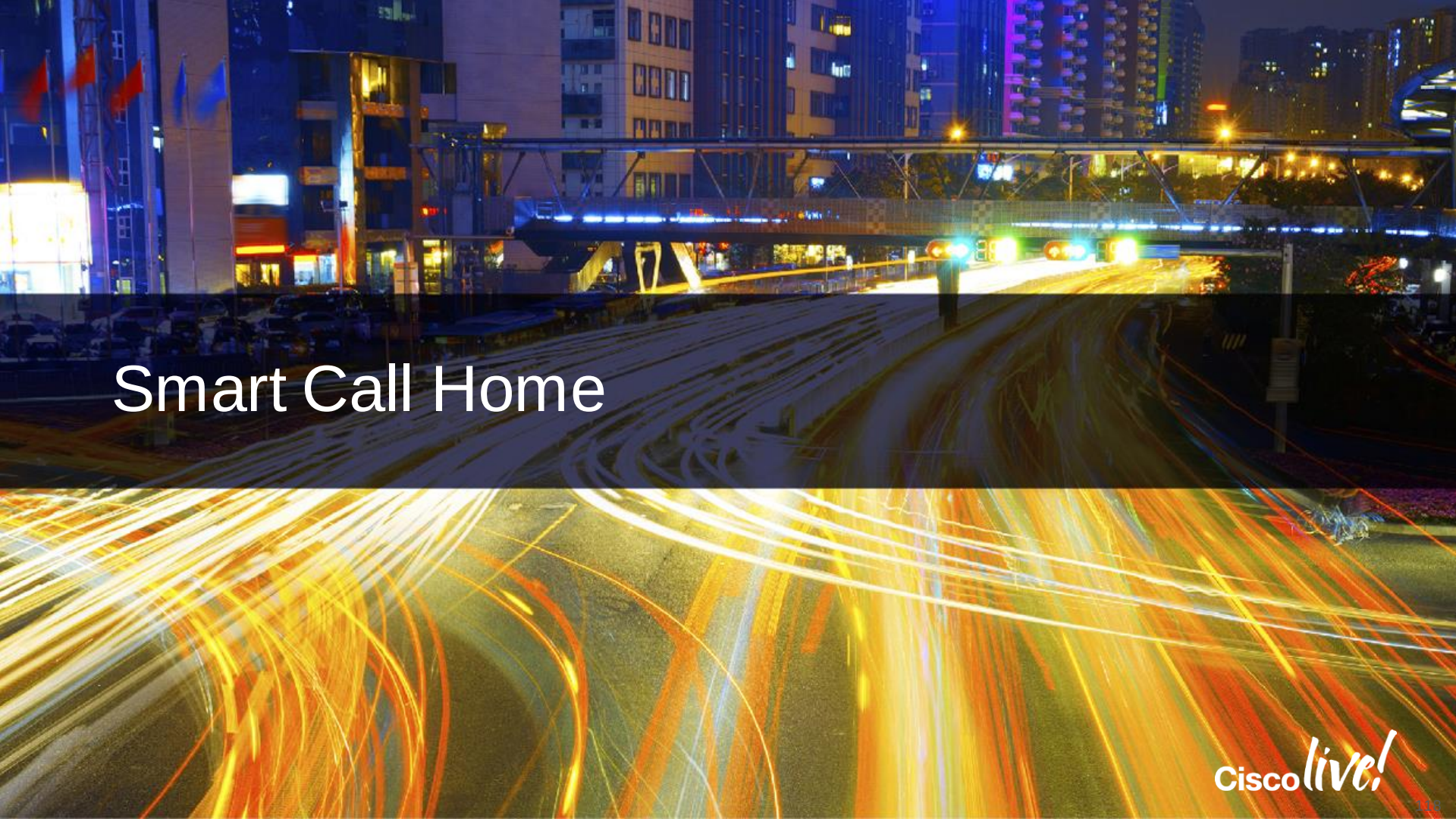
```
asa# ping tcp
 Interface: inside
 Target IP address: 72.163.4.161
 Target IP port: 80
 Specify source? [n]: y
 Source IP address: 192.168.1.101
 Source IP port: [0]
 Repeat count: [5]
 Timeout in seconds: [2]
 Type escape sequence to abort.
 Sending 5 TCP SYN requests to 72.163.4.161 port 80
 from 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




Smart Call Home

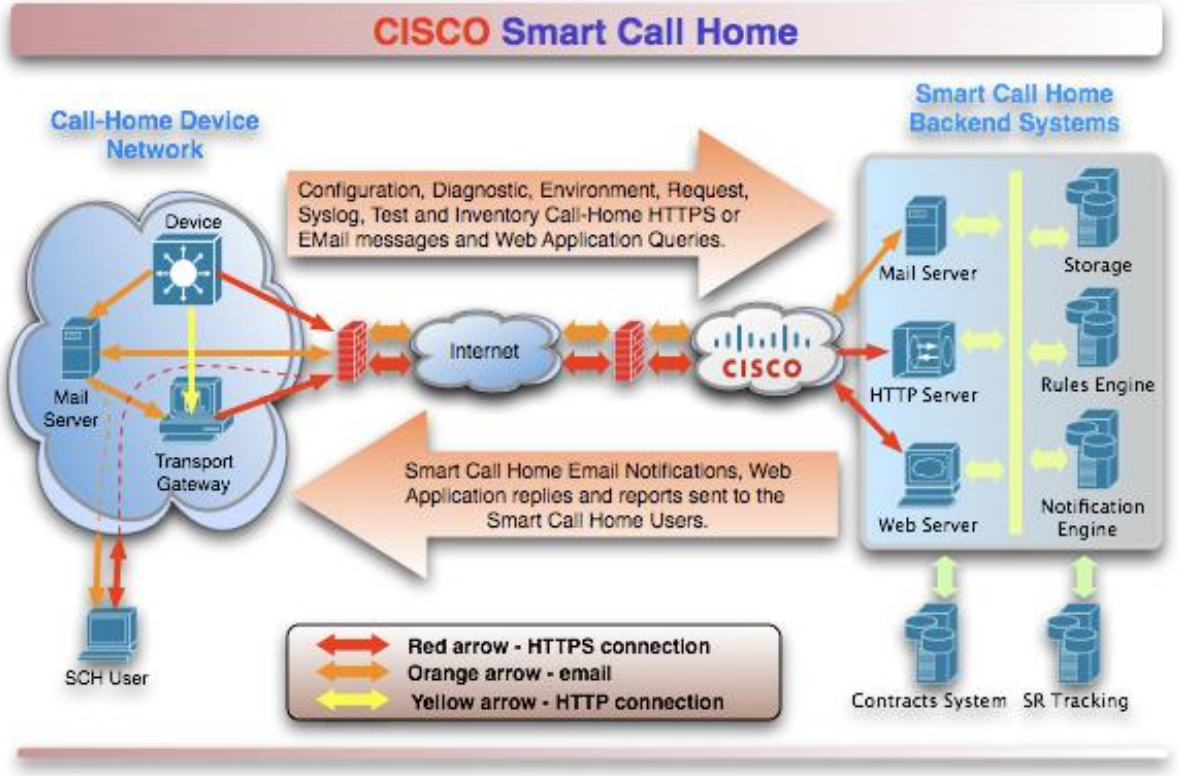
Smart Call Home

What is it?

- An embedded smart services capability.
- Automate communication between customer network and Cisco Technical services.
  - Proactive, real-time diagnostics and alerts.
  - Faster access to TAC by automatic generation of service requests.
  - Reporting capability using SCH portal.
- Built-in feature to many Cisco platforms.
- SCH on Cisco Support Community: https://supportforums.cisco.com/community/netpro/solutions/smart_services/smartcallhome
- SCH portal: https://tools.cisco.com/sch/

Smart Call Home Components

For your reference



Smart Call Home

What all can the ASA do?

- Many predefined possible messages grouped into **alert-groups**.

```
ciscoasa(cfg-call-home)# alert-group ?

call-home mode commands/options:
  all             Enable or disable all alert-groups
  configuration             Configuration Group
  diagnostic                Diagnostic Group
  environment               Environmental Group
  inventory                 Inventory Group
  snapshot                  Snapshot Group
  syslog          System Log Group
  telemetry                 Telemetry Group
  threat          Threat Group
```

- Ability to enable one or more alert-groups and send information to email or https destinations.

Smart Call Home
Configuration alert-group

- Triggered periodically, at system start-up or on-demand.
- Consists of output of the below show commands.
  - show access-list | in elements, show running-config, show startup-config, show call-home registered-module status | exclude disabled, show context.
- ASA in multi-context mode – one message each for system mode and for each context.
- 2 possible message options:
  - Full: includes all applicable command outputs above.
  - Minimum: includes only output of "show call-home registered-module status | ex disabled".

Smart Call Home

Diagnostic alert-group

- Triggered by diagnostic events.
- Classified into the following 3 message types.
  - Traceback: “show crashinfo” and “show tech-support no-config”
  - Mini-dump: Contents of system dump
  - Failover: “show failover” and “show failover history”
- Service request opened if no matching bug found for traceback messages sent to SCH server.
- E-mail notification sent for specific failover events (Interface check failure, service card failure, etc.)

Smart Call Home

Environment alert-group

- Triggered by environment events.
- Classified into the following 2 message types.
  - Health: fan, power, voltage and temperature monitoring (only supported platforms). Contains output of “show environment”.
  - Resource: cpu and memory monitoring. Contains “show cpu usage” and “show memory detailed”.
- Service request opened and/or e-mail notification for specific health events (fan failure, critical temperature, etc.).
- User-defined CPU and memory thresholds.

```
ciscoasa(cfg-call-home)# alert-group-config environment
ciscoasa(cfg-call-home-environment)# threshold ?
call-home-alert-group-environment mode commands/options:
  cpu     Configure cpu usage threshold
  memory  Configure memory usage threshold
```

Smart Call Home

Inventory alert-group

- Triggered periodically, on a failover event, on system start-up, or on-demand.
- Contains output of commands “show environment”, “show failover state”, “show inventory”, “show module” and “show version”.
  - Only contains system mode information in case of multiple mode.

Snapshot alert-group

- Triggered periodically or on-demand.
- Option to add any desired “show” command to the message.

Smart Call Home

Syslog alert-group

- Triggered by user defined syslog events which can be one of 2:
  - Set of syslog IDs.
  - A specific syslog severity level.
- Device buffers syslog events for 60 seconds. Message sent if there are any logs in the buffer and buffer is cleared up.

Telemetry alert-group

- Triggered periodically or on-demand.
- Contains following command outputs:

  show traffic, show conn count, show vpn-sessiondb summary, show vpn load-balancing, show local-host, show memory, show access-list | include elements, show interface, show phone-proxy media-sessions count, show phone-proxy secure-phones count, show threat-detection statistics protocol, show xlate count, show perfmon detail, show route

Smart Call Home

Threat alert-group

- Triggered by blocked host by botnet traffic filter or by a shunned host.
- Contains following command outputs:
  - show shun, show dynamic-filter reports top botnet-ports, show dynamic-filter reports top infected-hosts, show dynamic-filter statistics, show threat-detection rate, show threat-detection scanning-threat, show threat-detection shun, show threat-detection statistics top.

Configuration Procedure

- Enable the call-home service.
- Provide contact information.
- Enable the necessary alert-groups
- Create a profile.
- Optional configuration:
  - SMTP server for e-mail destinations.
  - DNS Server if names used in place of IP addresses.

```
service call-home
call-home
 contact-email-addr <email_addr>
 sender from <FROM: email_addr>
 sender reply-to <email_addr>
 mail-server <email_server> priority 1
 alert-group all
profile TAC
  active
  destination address email <TO: email_addr> msg-format long-
text
  destination transport-method email
  subscribe-to-alert-group diagnostic
```

SCH Examples - 1

Send a command output to a Service Request or E-mail address.

- Objective – To send the output of a command directly to an open service request.
- Let's take the example of the CLI command show run.

```
ciscoasa# call-home send "show run" service-number 6xxxxxxx
```

- Sends the output in plain-text format to attach@cisco.com with SR number in subject.
- Destination could also be an e-mail address:

```
ciscoasa# call-home send "show run" email abc@xyz.com
```

SCH Examples - 2

Periodic health monitoring of ASA

- Objective – To monitor relation of overruns with respect to CPU hogs on ASA every 5 minutes.
- We will leverage the snapshot alert-group.

```
service call-home
call-home
 alert-group-config  snapshot
  add-command  "show interface  | i Interface|overrun"
  add-command  "show process cpu-hog"
 contact-email-addr  abc@xyz.com
 sender from abc@xyz.com
 sender reply-to  abc@xyz.com
 mail-server  10.106.106.134 priority 1
 profile overrun-hog
  destination address email abc@xyz.com  msg-format  long-text
  destination transport-method  email
  email-subject  “Overruns  vs CPU Hogs"
  subscribe-to-alert-group  snapshot  periodic interval  5
```

- Any show command can be added to the snapshot.

SCH Examples - 2

Periodic health monitoring of ASA

- Example e-mail.

Overruns vs CPU Hogs

[redacted]

Sent: Sunday, 21 October 2012 1:58 PM

To: [redacted]

```
SCH Notification from asa5510-2-snapshot-none-2012-10-21 00:20:51 GMT+00:00

Time Stamp: 2012-10-21 00:20:51 GMT+00:00
Message Name: Snapshot
Message Type: snapshot
Severity Level: 10
Source ID: ASA5510
Device ID: [redacted]
Customer ID:
Contract ID:
Site ID:
Message Description: ASA Snapshot
Device Name: asa5510-2
Contact Name:
Contact Email: [redacted]
Contact Phone:
Street Address:
Affected Chassis: ASA5510
Affected Chassis Serial Number: [redacted]
Affected Chassis Hardware Version: 2.0
Affected Chassis Software Version: [redacted]
Command Output Name: show interface | i Interface|overrun
Command Output Text:
Interface Ethernet0/0 "", is administratively down, line protocol is up
        1981 input errors, 0 CRC, 0 frame, 1981 overrun, 0 ignored, 0 abort
Interface Ethernet0/1 "inside", is up, line protocol is up
        0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
Interface Ethernet0/2 "", is administratively down, line protocol is up
        0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
Interface Ethernet0/3 "", is administratively down, line protocol is up
        0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
Interface Management0/0 "management", is up, line protocol is up
        0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
Command Output Name: show process cpu-hog
Command Output Text:

CPU hog threshold (msec):  3.47
Last cleared: 00:04:36 UTC Oct 21 2012
```

SCH Examples - 3

Automatic notification when CPU/Memory goes high

- Objective – To receive an e-mail notification when CPU or memory goes high.
- We will leverage the environment alert-group.

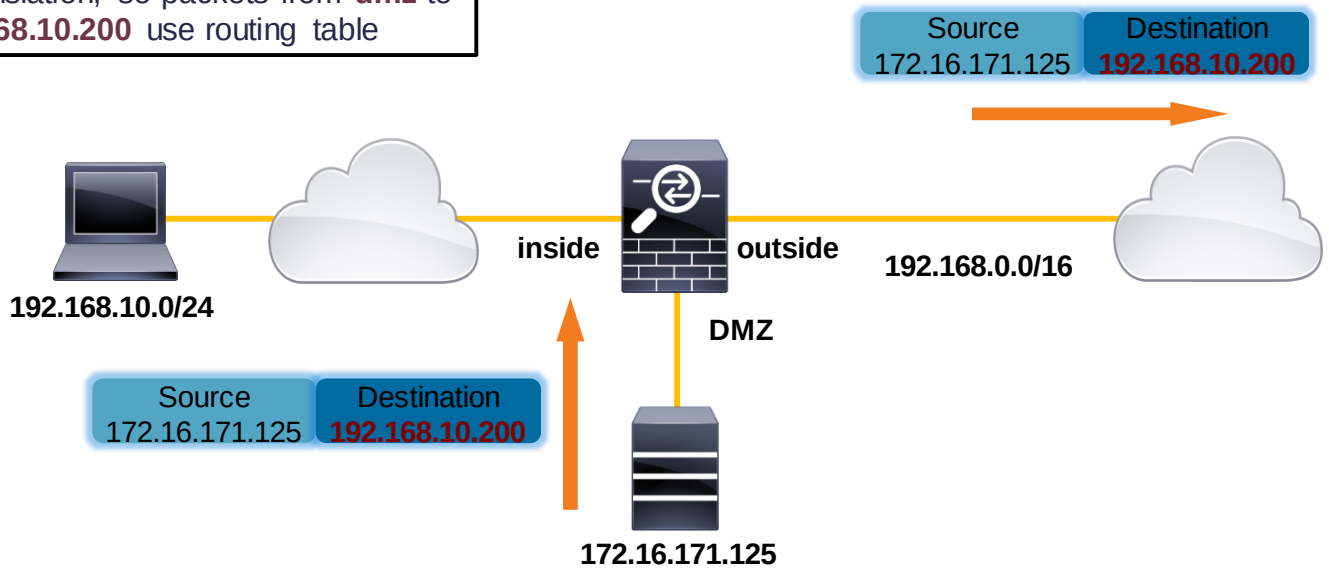
```
service call-home
call-home
 alert-group-config  environment
  threshold  cpu 70-90
  threshold  memory 60-70
 contact-email-addr  abc@xyz.com
 sender from abc@xyz.com
 sender reply-to  abc@xyz.com
 mail-server  10.106.106.134 priority 1
 profile cpu-mem
  destination address email abc@xyz.com  msg-format  long-text
  destination transport-method  email
  email-subject  “CPU/Memory  High"
  subscribe-to-alert-group  environment
```

NAT Traffic Diversion Examples

```
object network DMZ_FTP
 host 192.168.10.200
 nat (inside,dmz) static 192.168.10.200
```

Identity translation, so packets from **dmz** to **192.168.10.200** use routing table

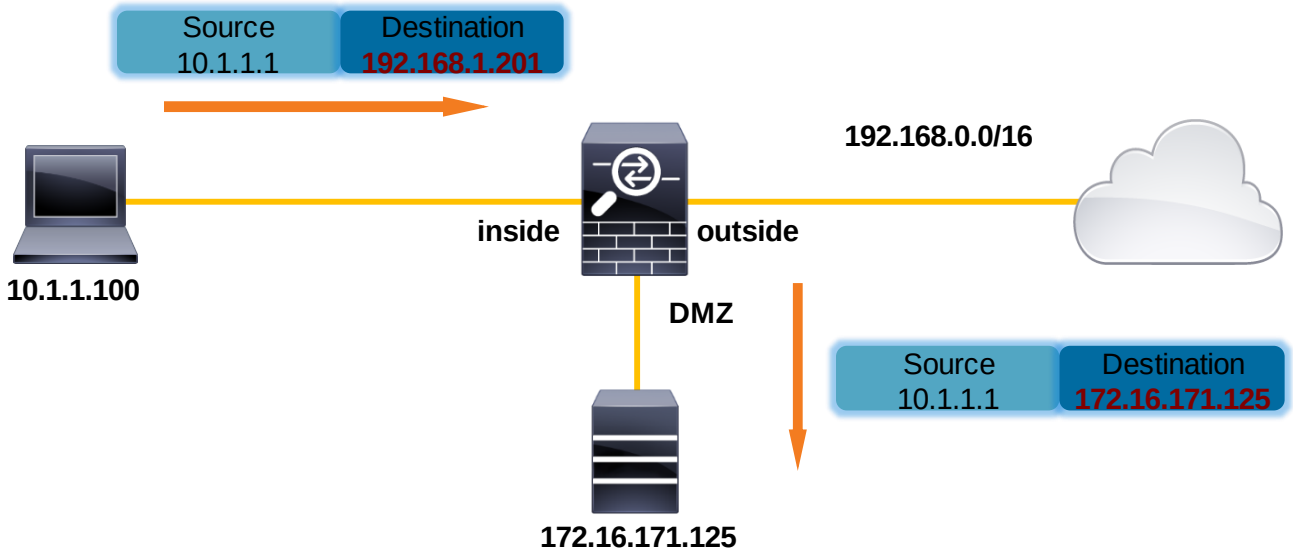
```
ciscoasa# sh route
S    0.0.0.0 0.0.0.0 [1/0] via 10.1.1.1, inside
C    192.168.0.0 255.255.0.0 is directly connected,
outside
```



NAT Traffic Diversion Example

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

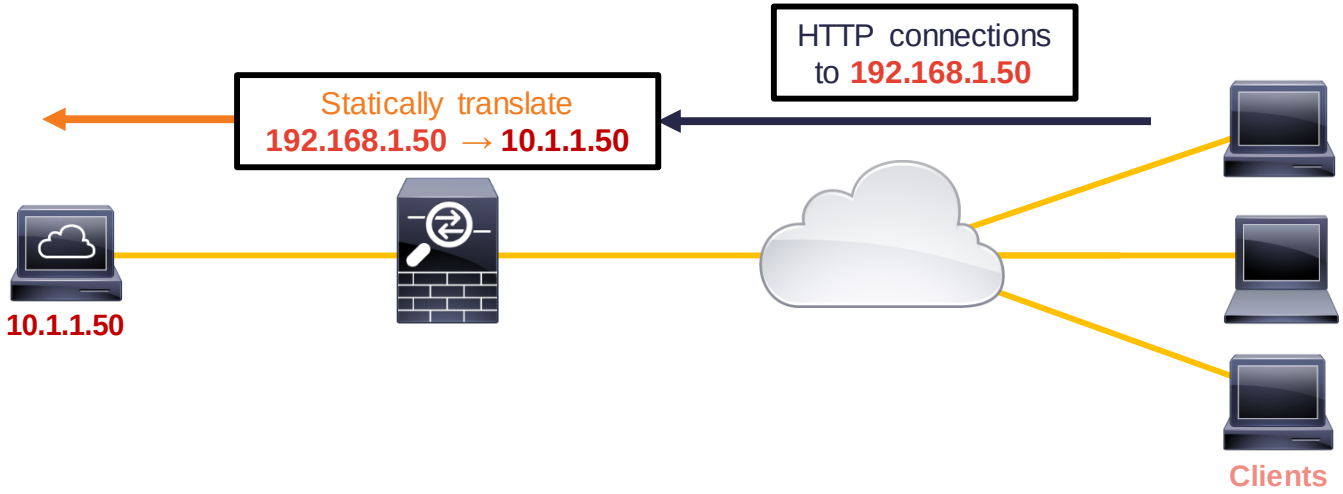
Actual translation, so inbound packets from **inside** to **192.168.1.201** will always divert to **172.16.171.125** on **DMZ**



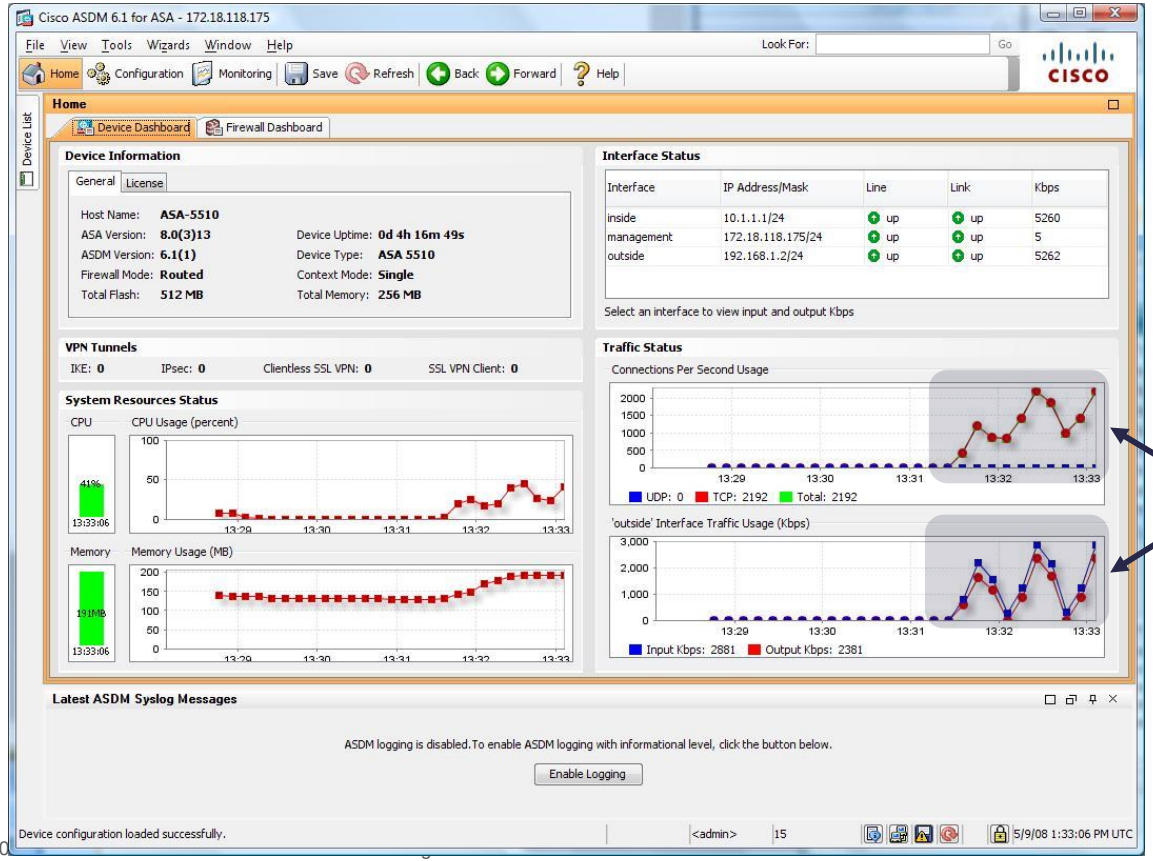
Case Study: Intermittent Access to Web Server

Problem Description

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



Monitoring Connection and Traffic Rates in ASDM



Huge connection and traffic spikes on outside interface

Checking Connection Rate Statistics

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

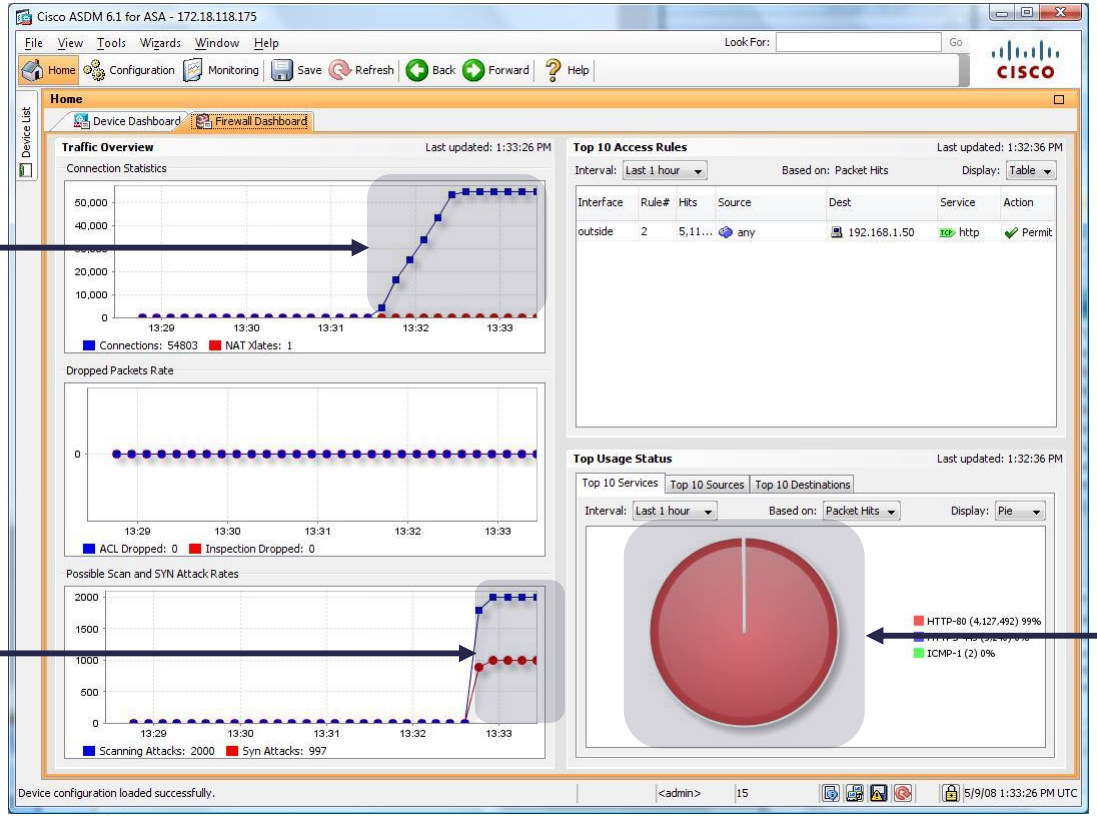
Current embryonic (half-open or incomplete) connection timeout rate is very high compared to the overall TCP connection rate

```
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%
```

Monitoring SYN Attack Rate in ASDM

Total connection count spikes

High level of incomplete connection attempts indicates a SYN flood attack



99% of connections is HTTP

Checking Incomplete TCP Connection Source

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

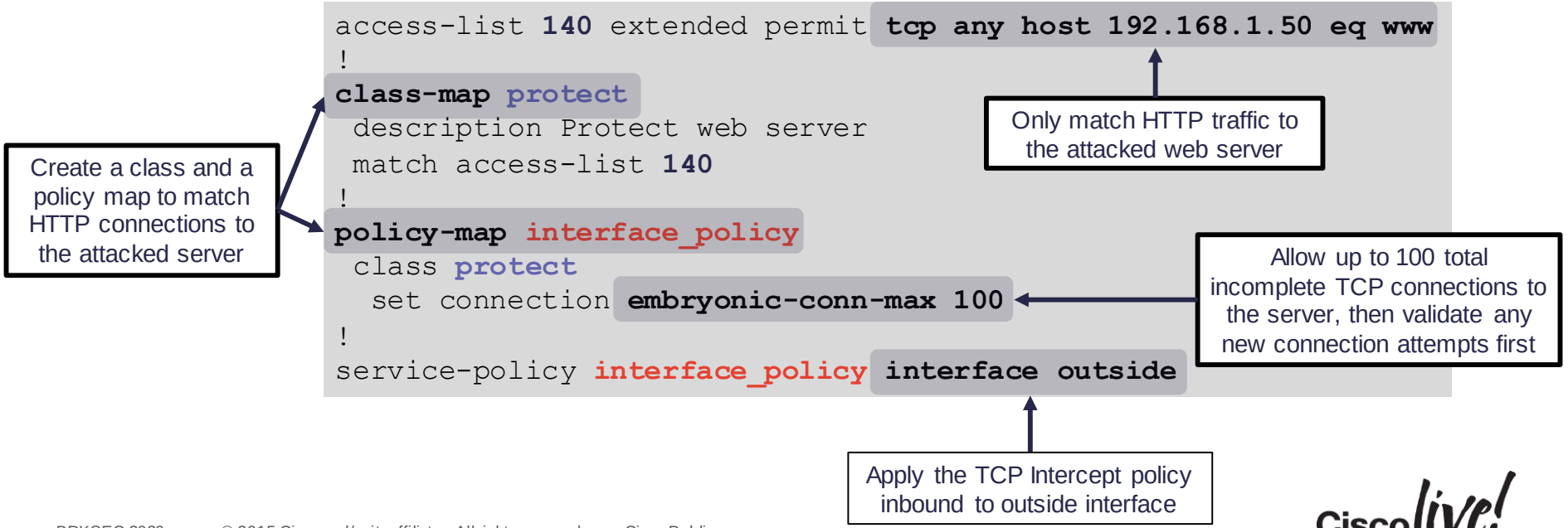
```
asa# show conn state tcp_embryonic
54764 in use, 54764 most used
TCP outside 17.24.101.118:26093 inside 10.1.1.50:80, idle 0:00:23, bytes 0, flags aB
TCP outside 111.76.36.109:23598 inside 10.1.1.50:80, idle 0:00:13, bytes 0, flags aB
TCP outside 24.185.110.202:32729 inside 10.1.1.50:80, idle 0:00:25, 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

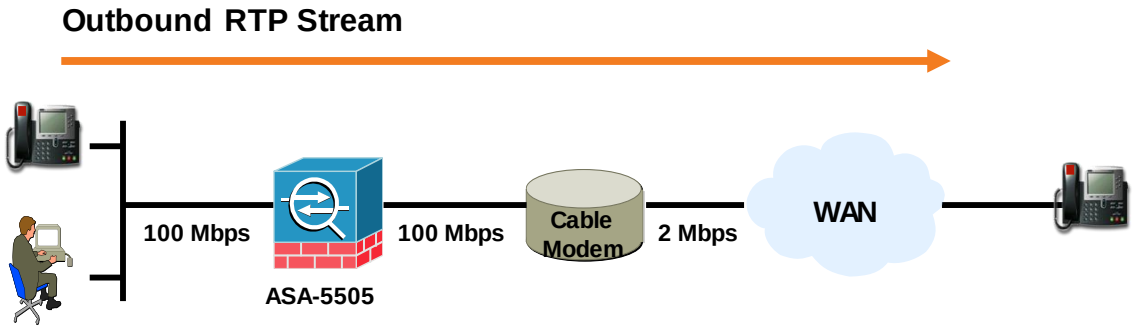


Case Study: Poor Voice Quality

Case Study: Poor Voice Quality

Problem

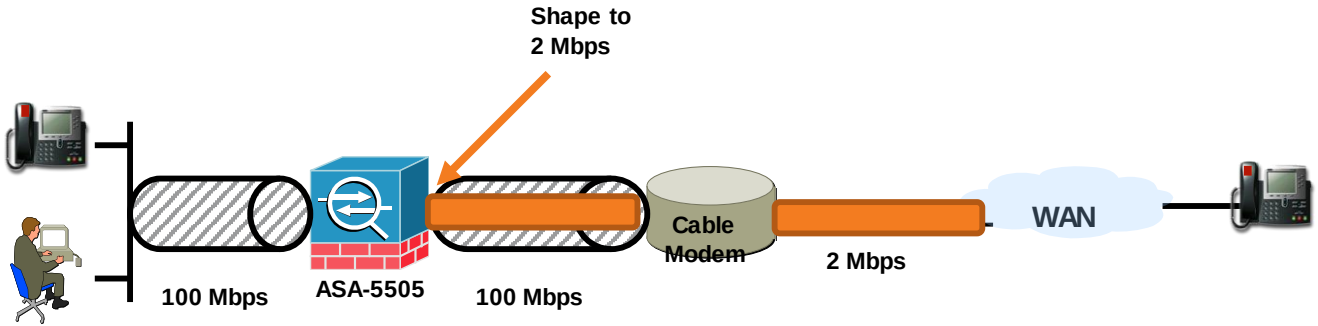
- Poor outbound voice quality at SOHO sites



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
```

FWSM

- Additional architecture information

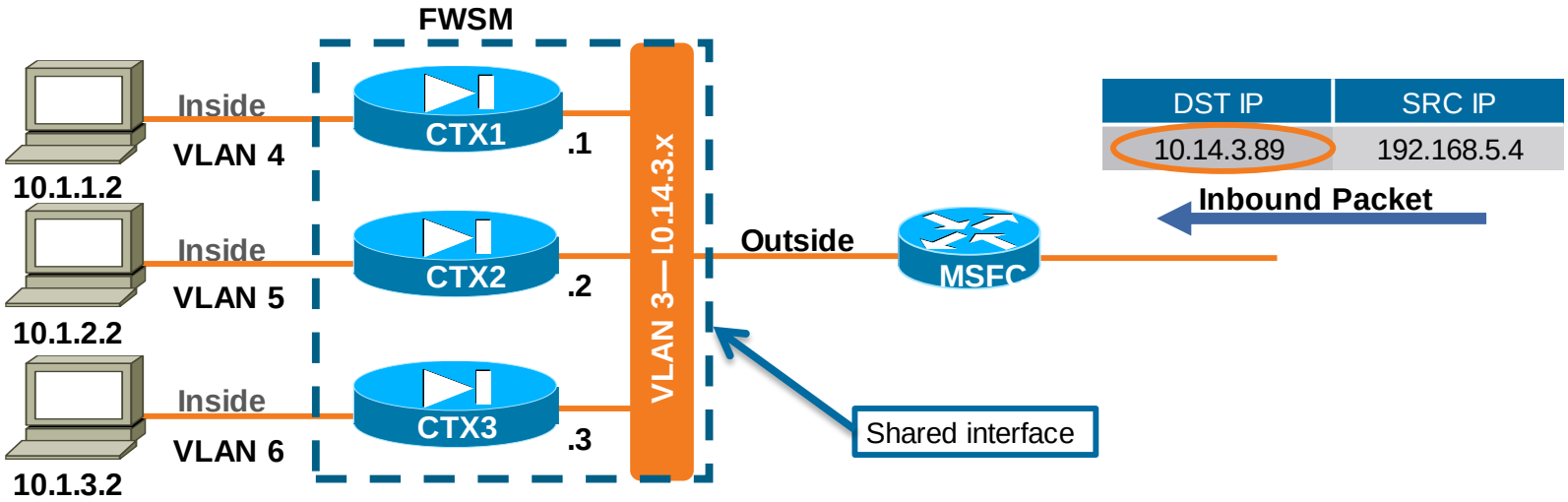
Classifier in Multimode

- When the firewall receives a packet, it must classify it to determine where to send the packet (which context)
- Packets are classified based on the following
  - Unique ingress interface/VLAN
  - Packet's destination IP matches a global IP
- FWSM has a single MAC address for all interfaces
- ASA has unique MAC per context for shared interfaces (physical interfaces have unique MACs)

Classifier in Multimode on FWSM

Example

- Inbound traffic is classified to context CTX3, based on the global IP in the NAT translation



Multi-Context - Common Issues on FWSM

- Overlapping statics (globals) across contexts
- Missing statics (globals), and unable to classify packets (shared inside interface) – check Admin context log

```
%FWSM-6-106025: Failed to determine security context for packet:
vlan3 tcp src 192.168.5.4/1025 dest 198.51.100.50/80
```

- Forgetting to 'monitor-interface' for Failover
- Forgetting to assign unique IP for each Transparent mode context
- Transparent mode, multi-BVI, one routing table

FWSM Syslog Level vs. Number of Messages

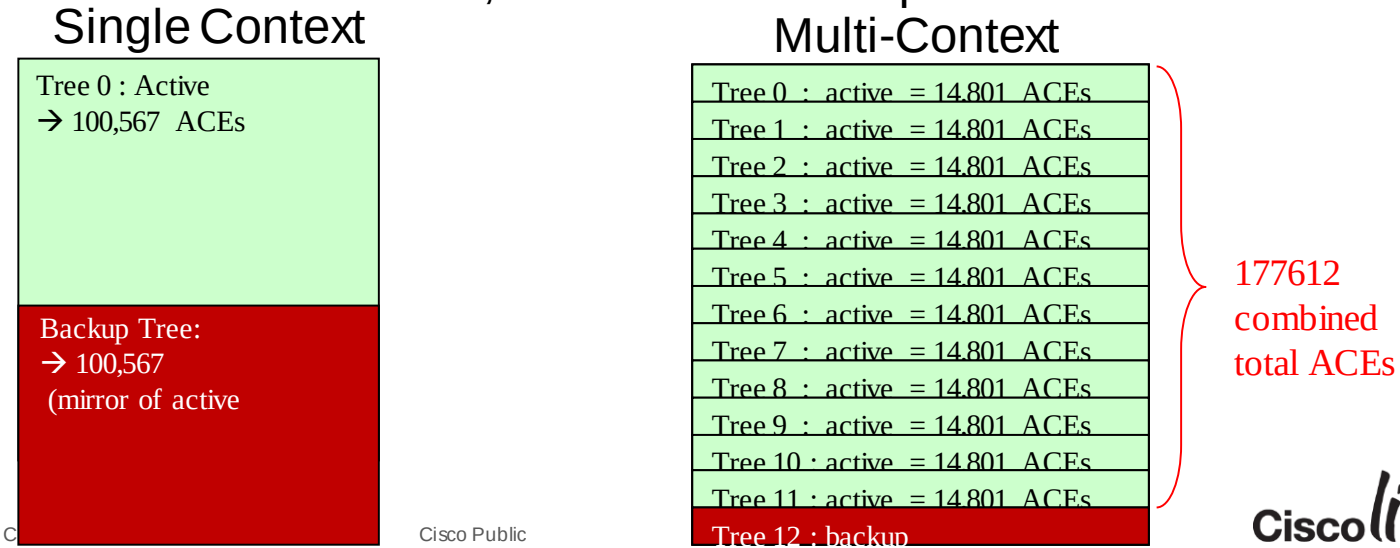
Log Level	Description	Number of Messages (SUM)				
		Ver. 2.3	Ver. 3.1	Ver. 3.2	Ver. 4.0	Ver. 4.1
0	Emergencies	0	0	0	0	0
1	Alerts	58 (58)	67 (67)	67 (67)	67 (67)	67 (67)
2	Critical	21 (79)	29 (96)	29 (96)	29 (96)	29 (96)
3	Errors	94 (173)	305 (401)	306 (402)	318 (414)	318 (414)
4	Warnings	131 (304)	194 (595)	196 (598)	199 (613)	199 (613)
5	Notifications	26 (330)	167 (762)	169 (767)	178 (791)	178 (791)
6	Informational	116 (446)	245 (1007)	248 (1015)	255 (1046)	259 (1050)
7	Debugging	23 (469)	225 (1232)	225 (1240)	226 (1272)	231 (1281)

FWSM and ACLs

- ACLs on the FWSM are compiled on the control point and pushed down into hardware (NP 3)
- During compile time, CPU should stay at ~ 99%
  - ACL compile uses all free CPU cycles
  - Allows compile to complete in shortest time possible
- Once compile is complete, rules are attempted to be pushed into hardware
  - Successful download
    - `Access Rules Download Complete: Memory Utilisation: 49%`
  - Failed download (exceeded HW memory)
    - `ERROR: Unable to add, access-list config limit reached`

FWSM - ACL Rule Limits

- ACL rules are about the only hardware limit users encounter
- In multimode, ACL resources are divided in 13 equal partitions (12 active, one backup)
- If you have less than 12 contexts, wasted reserved space



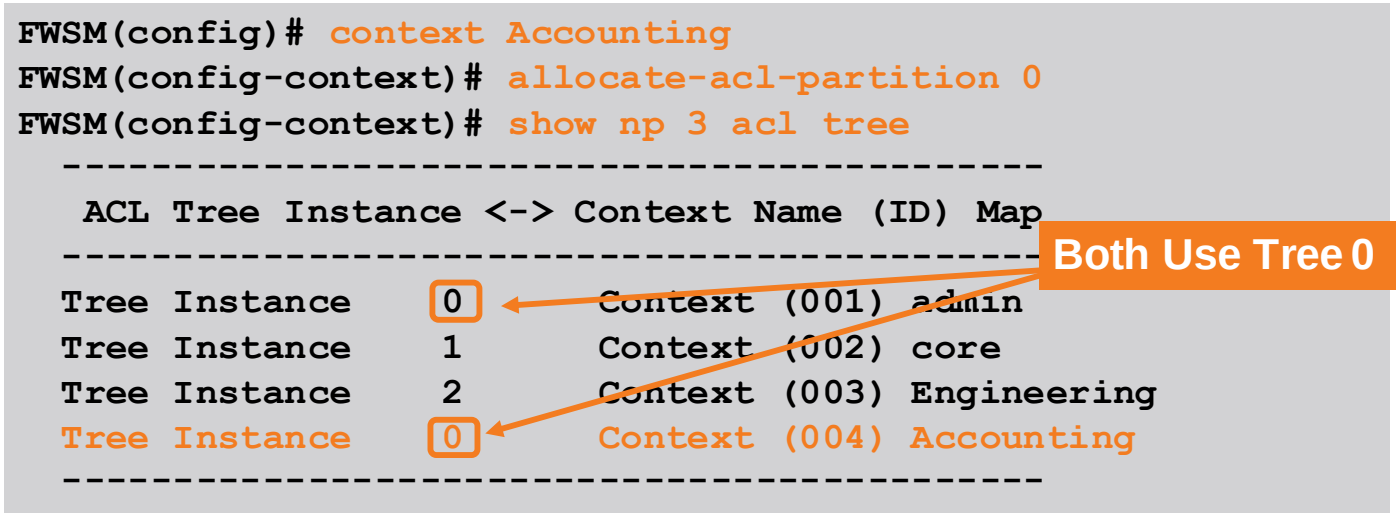
FWSM - ACL Rule Limits

- FWSM 2.3 introduced
  - `resource acl-partition`—set the number of ACL partitions
    `allocate-acl-partition`—assigns a context to a specific partition
- FWSM 3.2 introduced
  - `resource-rule`—allows further customisation of a partition
- FWSM 4.0 introduced
  - `resource partition`—customise the size of individual partitions
    `access-list optimisation enable`—merges and/or deletes redundant and conflicting ACEs without affecting the policy

FWSM and ACLs (Multimode)

- Use the command `resource acl-partition <num-of-partitions>` to reduce the number of active partitions created; default is 12
- Use the command `allocate-acl-partition <num>` to assign a context to a specific ACL tree

```
FWSM(config)# context Accounting
FWSM(config-context)# allocate-acl-partition 0
FWSM(config-context)# show np 3 acl tree
  ----------------------------------------
   ACL Tree Instance <-> Context Name (ID) Map
  ----------------------------------------
  Tree Instance    0      Context (001) admin
  Tree Instance    1      Context (002) core
  Tree Instance    2      Context (003) Engineering
  Tree Instance    0      Context (004) Accounting
  ----------------------------------------
```



FWSM - Resource Rule

- FWSM 3.2 introduced
  - **resource-rule**—allows further customisation of a partition

```
resource rule nat 10000 acl 2200 filter 400 fixup 595 est 70 aaa 555 console 283
```

  - **show resource-rule**—displays information about the current rule allocation

```
FWSM# show resource rule

             Default  Configured  Absolute
 CLS Rule     Limit      Limit      Max
-----------+---------+----------+---------
 Policy NAT    1843       1843      10000
 ACL          74188      74188      74188
 Filter        2764       2764       5528
 Fixup         4147       4147      10000
 Est Ctl        460        460        460
 Est Data       460        460        460
 AAA           6451       6451      10000
 Console       1843       1843       3686
-----------+---------+----------+---------
 Total        92156      92156

Partition Limit - Configured Limit = Available to allocate
     92156       -      92156         =           0
```

FWSM - Resource Partition

- FWSM 4.0 introduced
  - **resource partition**—allows customisation of the size of individual partitions (multi-context mode)

```
FWSM(config)# resource partition 10
FWSM(config-partition)# size 1000
WARNING: The rule max has been reset based on partition size 1000.
The <size> command leads to re-partitioning of ACL Memory.
It will not take effect until you save the configuration and reboot.
```

Before

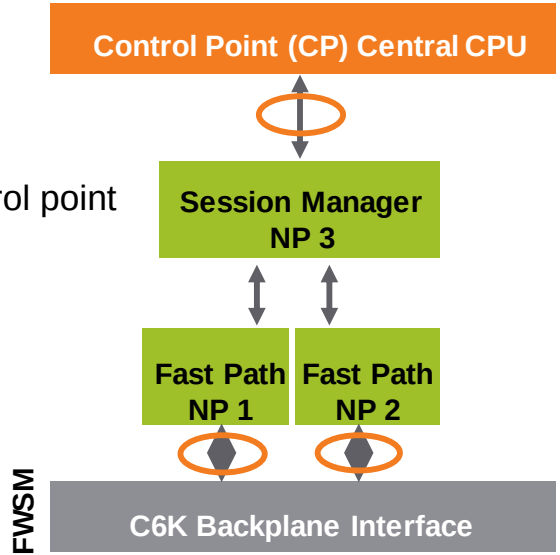
```
FWSM# show resource rule partition 10

             Default  Configured  Absolute
 CLS Rule     Limit      Limit      Max
-----------+---------+----------+---------
 Policy NAT     384        384       833
 ACL          14801      14801     14801
 Filter         576        576      1152
 Fixup         1537       1537      3074
 Est Ctl         96         96        96
 Est Data        96         96        96
 AAA           1345       1345      2690
 Console        384        384       768
-----------+---------+----------+---------
 Total        19219      19219

Partition Limit - Configured Limit = Available to
allocate
      19219     -      19219       =           0
```

After

```
FWSM# show resource rule partition 10

             Default  Configured  Absolute
 CLS Rule     Limit      Limit      Max
-----------+---------+----------+---------
 Policy NAT      20         20        43
 ACL            770        770       770
 Filter          30         30        60
 Fixup           80         80       160
 Est Ctl          5          5         5
 Est Data         5          5         5
 AAA             70         70       140
 Console         20         20        40
-----------+---------+----------+---------
 Total         1000       1000

Partition Limit - Configured Limit = Available to allocate
       1000     -       1000       =           0
```

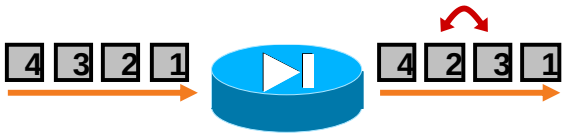
FWSM and Control Point

- The traffic that makes it to the control point is traffic that requires Layer 7 fixup (embedded NAT, or cmd inspection)
  - FTP
  - VoIP (SIP/SKINNY/H.323/RTSP)
  - DNS
  - XDMCP, etc.
- Traffic sourced from, or destined to, the FWSM also goes through the control point
  - Syslogs
  - AAA (RADIUS/TACACS+)
  - URL filtering (WebSense/N2H2)
  - Management traffic (telnet/SSH/HTTPS/SNMP)
  - Failover communications
  - Routing protocols (OSPF/ RIP)
  - etc.



FWSM - Enabling the Completion Unit

- Due to the FWSM's NP architecture, there exists a possibility that packets arriving with a low inter-packet gap might be re-ordered by the firewall



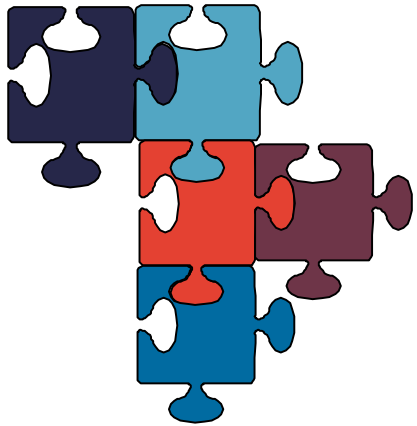
- This issue might be encountered when performing TCP throughput testing, or passing high speed TCP flows through the FWSM
  - Examples: CIFS, FTP, AFP, backups
- FWSM version 3.1(10) and 3.2(5) introduce a new command **`sysopt np completion-unit`** to ensure the firewall maintains the packet order (by enabling a hardware knob on the NPs called the completion unit)
- In multiple mode enter this command in the admin context configuration; It will then be enabled for all contexts on the firewall

Case Study – Advanced Syslog Analysis

Cisco live!

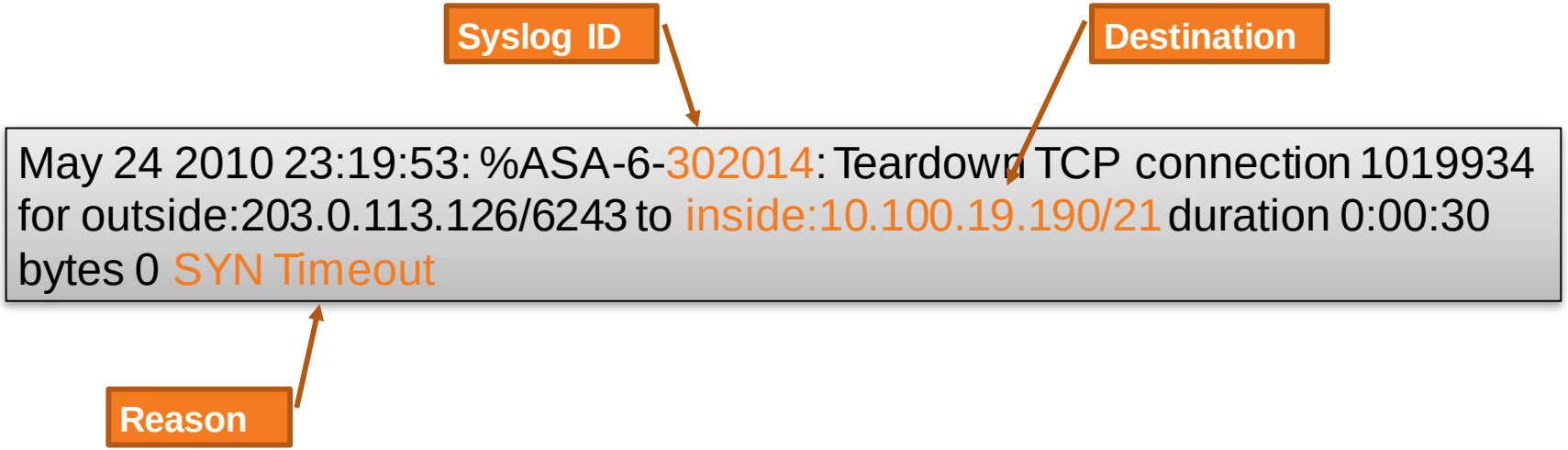
Case Study: Advanced Syslog Analysis

- 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:



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
```

Failover

- What to Do After a Failover
- Additional Failover Commands

What to Do After a Failover

- Starting with FWSM 2.3 and Cisco ASA/PIX 7.0, the reason for failover is saved in the failover history
- This information is not saved across reboots

```
ASA# show failover history
==========================================================================
From State                  To State                    Reason
==========================================================================
Disabled                    Negotiation                 Set by the CI config cmd
Negotiation                 Just Active                 No Active unit found
Just Active                 Active Drain                No Active unit found
Active Drain                Active Applying Config      No Active unit found
Active Applying Config      Active Config Applied       No Active unit found
Active Config Applied       Active                      No Active unit found
Active                      Failed                      Interface check
==========================================================================
```

Other Useful Failover Commands

- **`failover exec mate`** – allows you to execute commands on the peer and receive the response back.
- **`failover reload-standby`** – only valid on Active unit
- **`prompt`** – changes the prompt to display failover priority and state.

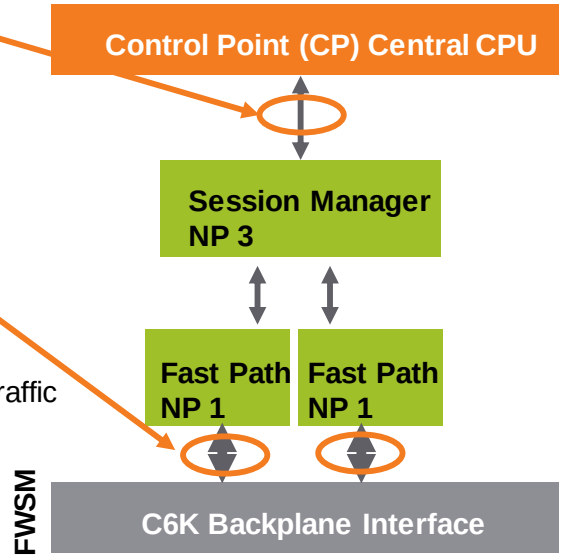
```
ASA(config)# prompt hostname priority state
ASA/sec/act(config)#
```

Failover Prompt Display Configuration

- The firewall's prompt maybe changed to display certain keyword
- Usage
  - prompt <keyword> [<keyword> ...]
- Syntax
  - keywords:
  - Hostname Configures the prompt to display the hostname
  - Domain Configures the prompt to display the domain
  - Context Configures the prompt to display the current context (multi-mode only)
  - Priority Configures the prompt to display the failover lan unit setting
  - State Configures the prompt to display the current traffic handling state
  - Slot Configures the prompt to display the slot location (when applicable)
- Example
  - `FWSM(config)# prompt hostname domain priority state slot`
  - `FWSM/cisco.com/sec/actNoFailover/4(config)#`

Packet Capture: Limitations on FWSM

- Capture functionality is available on the FWSM starting in 2.3
  - –However, only packets processed by the control point could be captured
- FWSM 3.1(1) added support to capture packets in hardware
  - –Only ingress packets were captured
- FWSM 3.1(5) both ingress and egress transient packets can be captured which flow through hardware
  - –Capture requires an ACL to be applied
  - –Capture copies the matched packets in hardware to the control point where they are captured; be careful not to flood the control point with too much traffic




Online Tools

TAC Security Podcast

Knowledge from TAC… On the go!

- Monthly podcast episodes with troubleshooting tips from TAC
- Focus on Cisco Security Technologies like ASA Clustering, Anyconnect, ISE, Voice Security, etc…and CCIE study tips!

40+ Episodes already available!

Cisco Technical Support App

FOLLOW US ON twitter

@CiscoTACPodcast

Cisco live!

Podcast Episodes

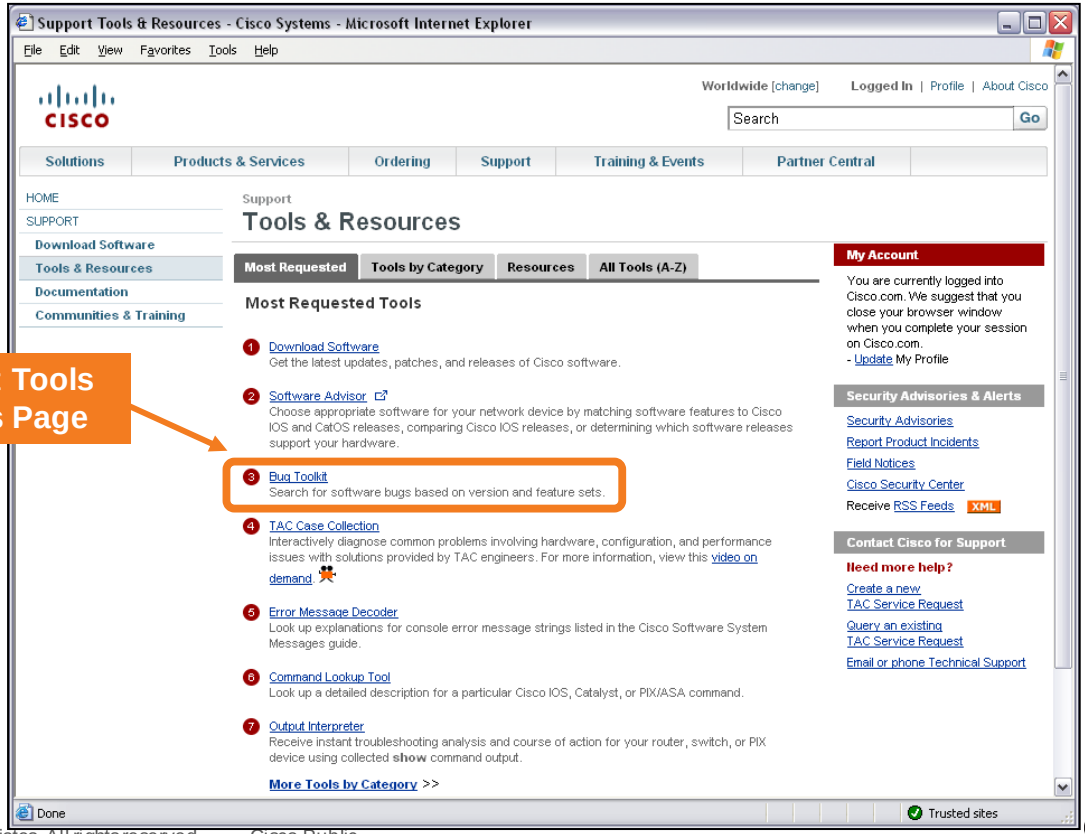
Ep. #	Topic	Ep. #	Topic
41	Troubleshooting ASA Clustering	27	IOS Embedded Event Manager (EEM)
40	Introduction to ASA Clustering	26	Troubleshooting IPSec VPNs
39	Voice Security Concepts and Best Practices	25	Understanding DMVPN and GETVPN
38	Introduction to OnePK	24	The Cisco Identity Services Engine
37	ASA Network Address Translation (NAT)	23	The Cisco ASA Services Module
36	Network Management at Cisco Live! 2013	22	How Cisco uses the Web Security Appliance to protect its network
35	Identity Services Engine v1.2	21	Cisco Live! Las Vegas 2011
34	Cisco Live! 2013 Orland, FL	20	This Week In TAC!
33	Virtual Security: The ASA 1000v and Virtual Security Gateway (VSG)	19	Troubleshooting the NAC Appliance
32	Investigating Syslogs: Tips and Tricks	18	Useful ASA and IPS Commands and Features You Might Not Know About
31	A look into ASA Quality with the Quality Assurance Team	17	Answering Questions From The Cisco Support Community
30	Introducing FlexVPN	16	Mitigating a SQL attack with ASA, IPS and IOS Firewall
29	Cisco Live! 2012 San Diego	15	Using Certificates on the ASA and IOS platforms
28	The History of the PIX	14	TCP connections through the ASA and FWSM

Podcast Episodes

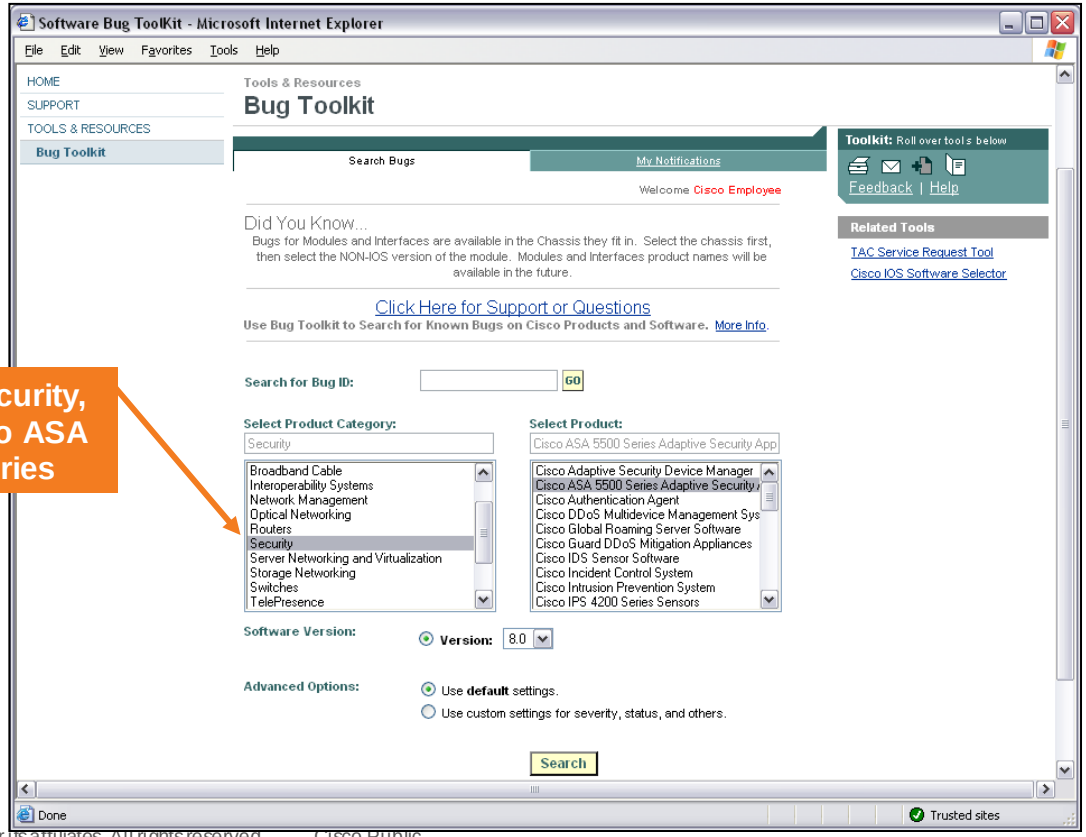
Ep. #	Topic
13	HTTP Filtering on the ASA
12	Securing Cisco Routers
11	ASA Anyconnect VPN
10	ASA Version 8.3 Overview
9	Multiple Context Mode on the ASA and FWSM Platforms
8	ASA Advanced Application Protocol Inspection
7	Monitoring Firewall Performance
6	Tips for Taking the CCIE Security Exam
5	Troubleshooting Firewall Failover, Part 2
4	Troubleshooting Firewall Failover Part 1; Guest Omar Santos from PSIRT
3	Transparent Firewall Mode; Lifecycle of a TAC Case
2	New Features Introduced with ASA Version 8.2
1	Using the ASA Packet Capture Utility for Troubleshooting

Bug Toolkit

On the Support Tools and Resources Page

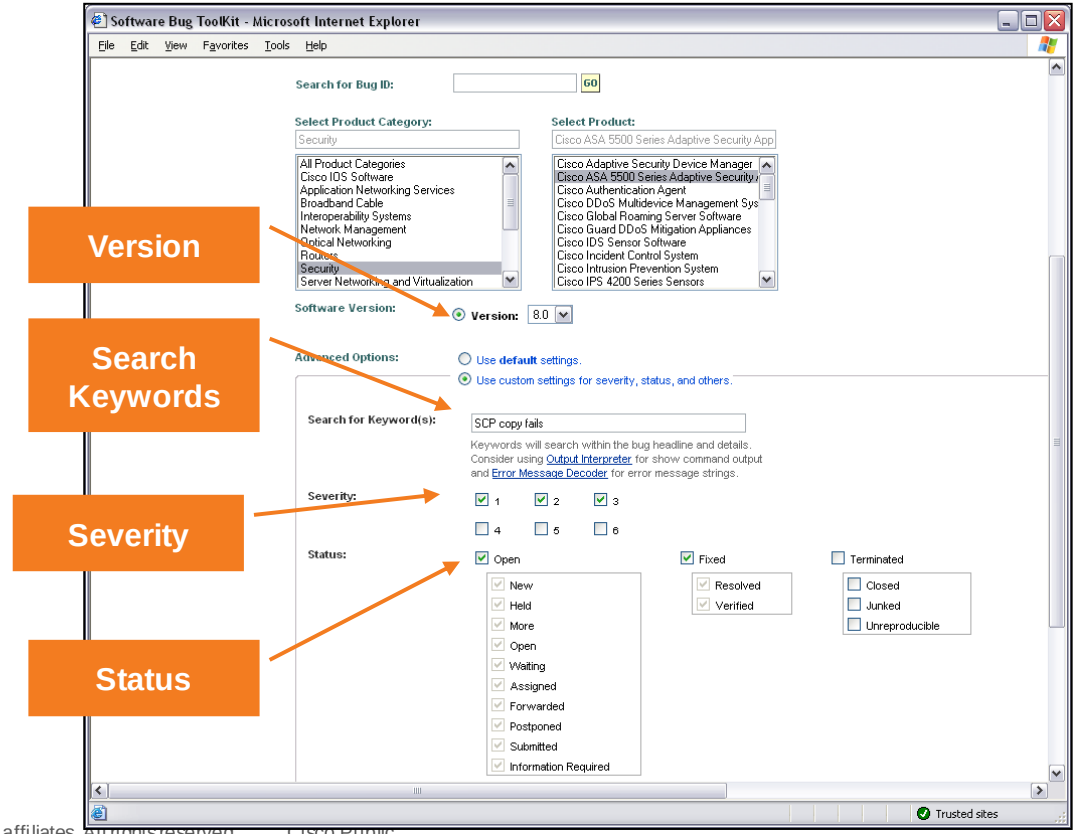


Bug Toolkit - Product Selection

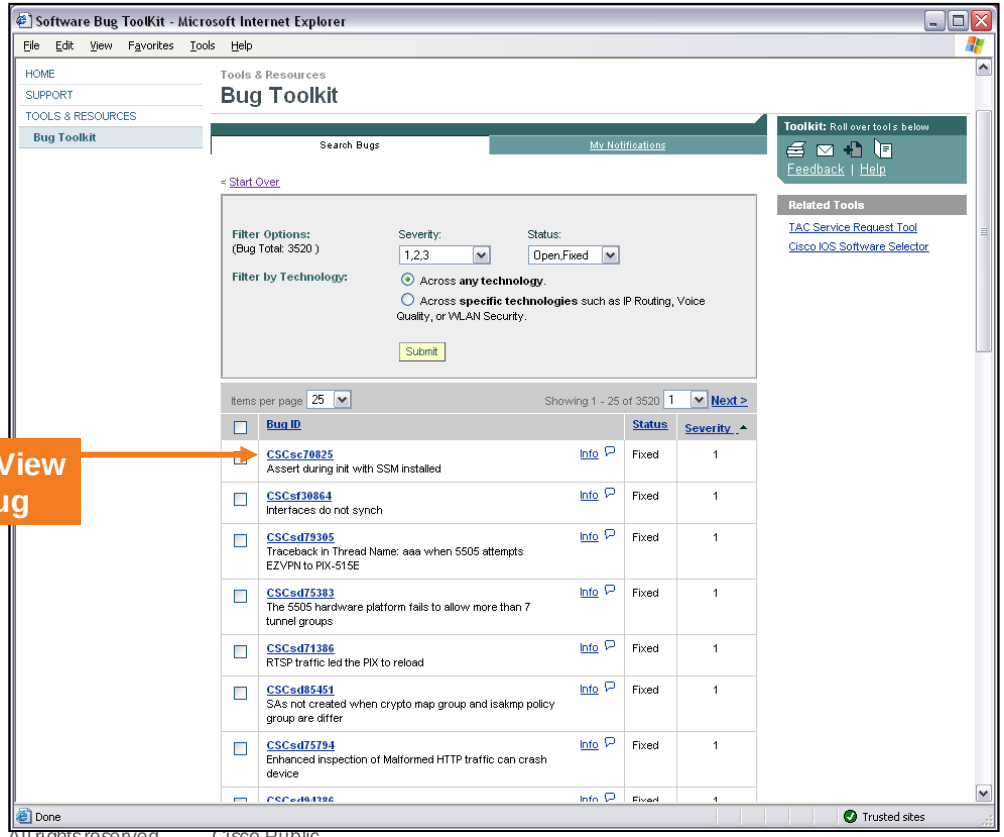


Select Security, then Cisco ASA 5500 Series

Bug Toolkit - Advanced Search

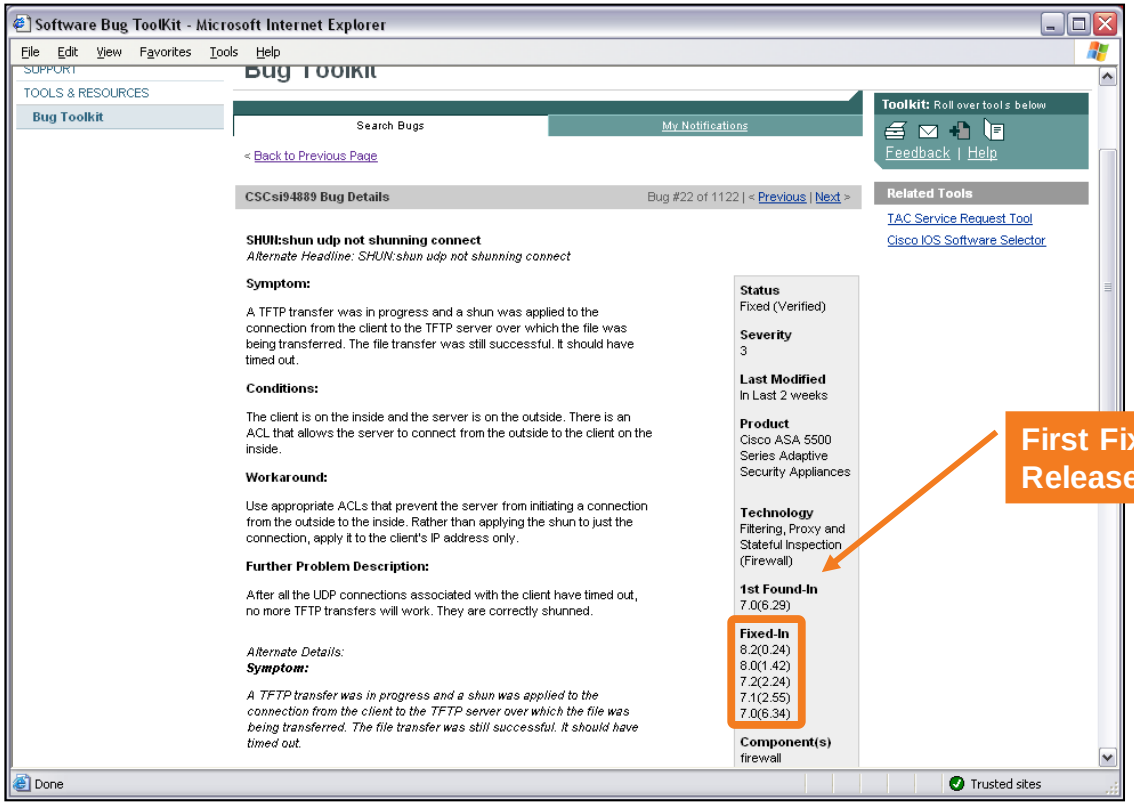


Bug Toolkit - Search Results



Select Link to View Details of Bug

Bug Toolkit - Bug Details

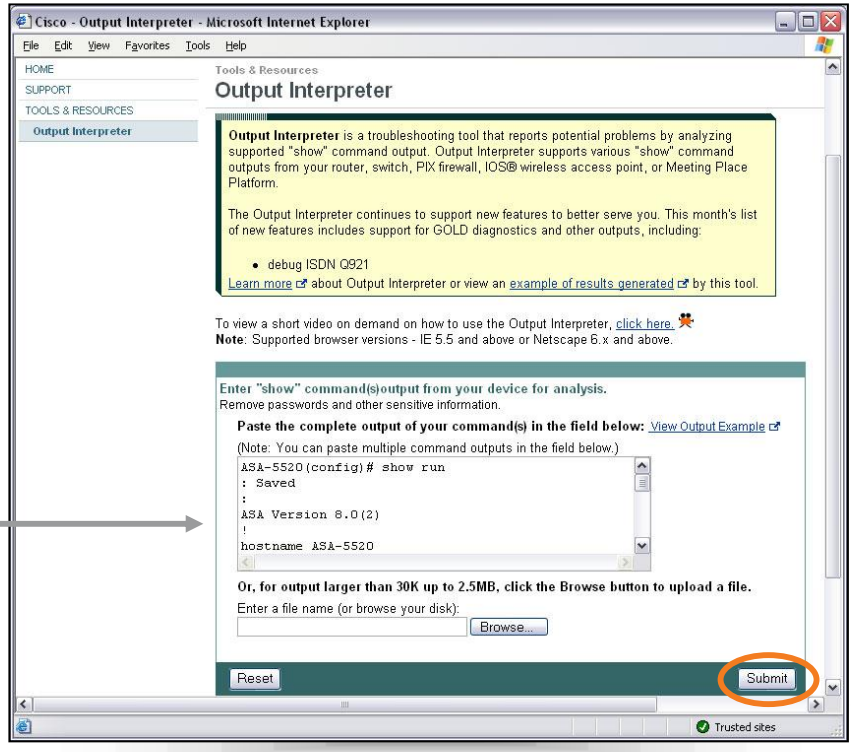


Output Interpreter

Linked off the Technical Support and Documentation—Tools and Resources Section on CCO

Great Tool for Catching Configuration Errors

Paste in the show run Output and Hit submit



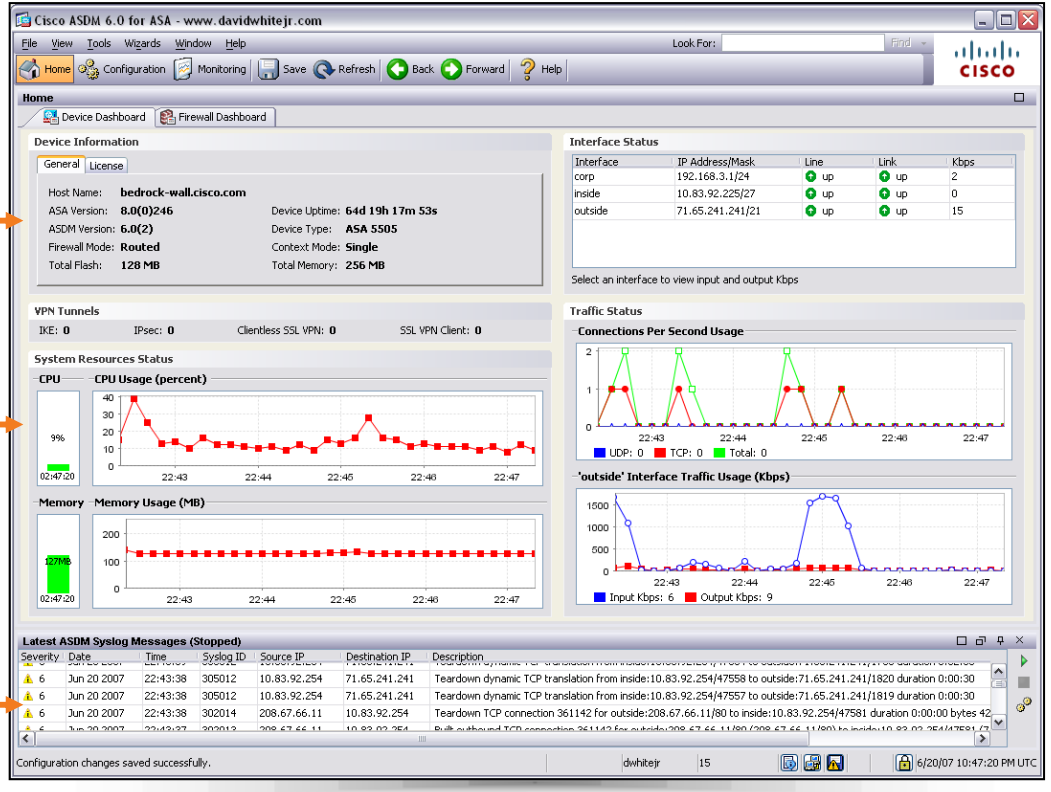
ASDM

ASDM Home Page

Device Information

CPU, Memory, Conns/Sec, Interface Traffic

Real-Time Syslogs

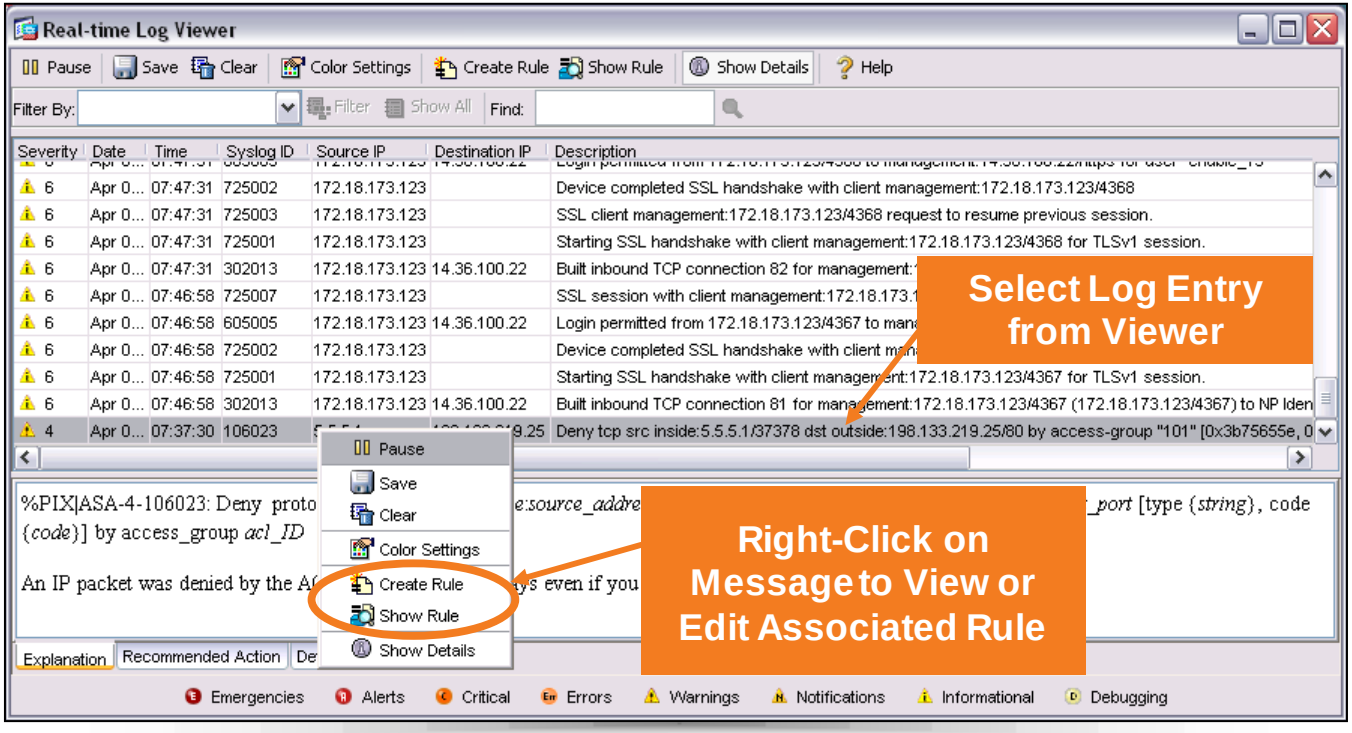


Using ASDM for Monitoring

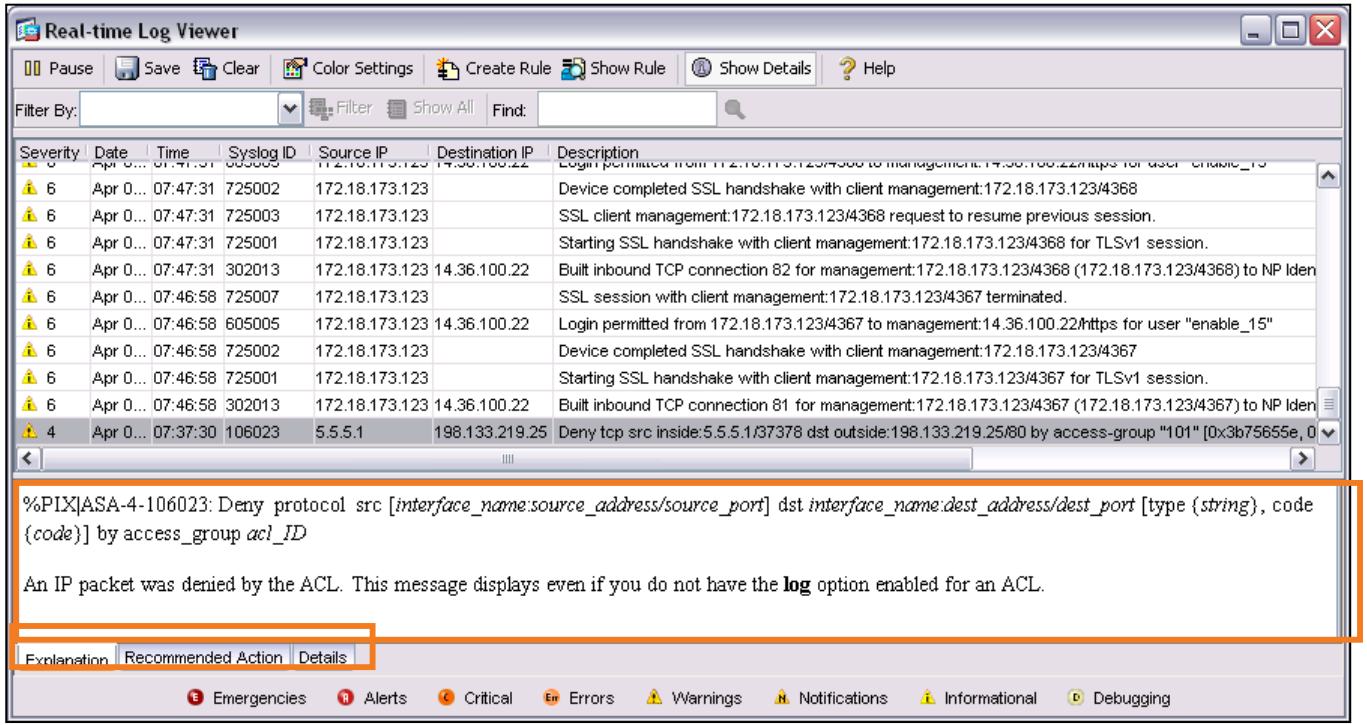
Great for Monitoring Trends

Up to Four Different Graphs Can Be Displayed

ASDM: Editing Rules from the Log Viewer



ASDM: Syslogs Explained



Opening a TAC Case

- If after using all your troubleshooting tools you still cannot resolve the problem, please open a TAC case
  - http://www.cisco.com/techsupport/servicerequest/
- At a minimum include:
  - Detailed problem description
  - Output from **show tech**
- Optionally include:
  - Syslogs captured during time of problem
  - Sniffer traces from both interfaces using the **capture** command (capturing only the relevant packets, and saved in pcap format)

Supportforums.cisco.com

- Ask a Question (for Free!)
- Hundreds of Sample Configs
- Troubleshooting Docs
- FAQs

http://supportforums.cisco.com/

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


CISCO