LET’S BUILD TOMORROW TODAY
Maximizing Firewall Performance
Andrew Ossipov, Technical Marketing Engineer
BRKSEC-3021
Your Speaker

Andrew Ossipov

aeo@cisco.com

Technical Marketing Engineer

8 years in Cisco TAC

18+ years in Networking
Cisco ASA Sessions: Building Blocks

BRKSEC-2020
Firewall Deployment
(Mon 10:00)

BRKSEC-2021
Firewall Architectures
(Wed 08:00)

BRKSEC-3032
ASA Clustering Deep Dive
(Th 10:00)

BRKSEC-3021
Maximizing Firewall Performance
(Tue 08:00)

BRKSEC-2028
Deploying NG Firewall
(Mon 08:00)
(Th 13:00)
Agenda

- Performance at a Glance
- Firewall Architecture
- Data Link Layer
- Connection Processing
- Transport Protocols
- Application Inspection
- Closing Remarks
Performance at a Glance
Defining Network Performance

• Throughput
  • Bits/sec, packets/sec
  • File transfers, backups, database transactions

• Scalability
  • New conns/sec, concurrent conns
  • Web, mobile users, VPN

• Reliability
  • Latency, jitter, packet loss
  • Real time applications, voice, video
Testing Performance

• Maximum throughput and scalability with UDP
  • Sufficient number of flows for proper load-balancing
  • Packet size: maximum for bytes/sec, minimum for packets/sec
  • Minimum features

• “Real World” profile is most trustworthy
  • Single (440-byte TCP) or multi-protocol (weighted mix)
  • Traffic patterns of an “average” network
## Reading Data Sheets

<table>
<thead>
<tr>
<th></th>
<th>ASA 5506-X</th>
<th>ASA5545-X</th>
<th>ASA5585 SSP40</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max Throughput</td>
<td>750Mbps</td>
<td>3Gbps</td>
<td><strong>20Gbps</strong></td>
</tr>
<tr>
<td>Real-World Throughput</td>
<td>300Mbps</td>
<td>1.5Gbps</td>
<td><strong>10Gbps</strong></td>
</tr>
<tr>
<td>Max AVC+NGIPS Throughput</td>
<td>125Mbps</td>
<td>1Gbps</td>
<td><strong>6Gbps</strong> (w/SFR-40)</td>
</tr>
<tr>
<td>Max VPN Throughput</td>
<td>100Mbps</td>
<td>400Mbps</td>
<td><strong>3Gbps</strong></td>
</tr>
<tr>
<td>64 Byte Packets/sec</td>
<td>-</td>
<td>900,000</td>
<td><strong>6,000,000</strong></td>
</tr>
<tr>
<td>Max Conns</td>
<td>50,000</td>
<td>750,000</td>
<td><strong>4,000,000</strong></td>
</tr>
<tr>
<td>Max Conns/sec</td>
<td>5,000</td>
<td>30,000</td>
<td><strong>200,000</strong></td>
</tr>
<tr>
<td>IPSEC VPN Peers</td>
<td>50</td>
<td>2500</td>
<td><strong>10,000</strong></td>
</tr>
<tr>
<td>Max Interfaces</td>
<td>8x1GE</td>
<td>14x1GE</td>
<td><strong>6x1GE + 20x10GE</strong></td>
</tr>
</tbody>
</table>

### Key Notes:
- **Max > Real-world > AVC+NGIPS > VPN**
- **64 bytes x 8 bits/byte x 6M packets/sec = 3.07Gbps**
- **4,000,000 conns/200,000 conns/sec = 20 seconds**
- **3Gbps/10,000 peers = 300Kbps/peer**
- **206Gbps >> Max**
Firewall Capacities

- **Interface bound**
  - Line rate, packet rate, throughput
  - Load-balancing matters

- **CPU bound**
  - Conn setup rate, throughput, features
  - Back pressure on interfaces and network

- **Memory bound**
  - Maximum conns, policy rules, throughput
  - Utilization affects entire system

- **External delays beyond firewall**
  - Latency, conn setup rate, throughput
ASA Architecture
ASA5505 Block Diagram

- **CPU**
- **Crypto Engine**
- **RAM**
- **Expansion Slot IPS SSC**
- **Internal Switch**
  - 8x100Mbps
- **External Switched Ports**
  - 8xFE

Connections:
- 1Gbps to Internal Switch
- System Bus
- Ethernet
ASA5506-X, 5508-X, and 5516-X Block Diagram

- External Interfaces: 8x1GE
- Management 1/1: 1GE
- Network Processor Unit (NPU) and Crypto Engine
- SFR/IPS/CX CPU
- Firewall CPU
- Firewall RAM
- NPU RAM
- Wireless ASA5506W-X
- External NICs: 8x1Gbps
- System Bus
- Ethernet
ASA5500-X Block Diagram

- **SFR/IPS/CX CPU**
- **External Interfaces**
  - 6x1GE
- **On-board Interfaces**
  - 6x1GE* or 8x1GE**
- **Crypto Engine**
- **Management0/0 1GE**
- **Bus 0**
- **Bus 1**
- **Expansion Card**
  - 6x1Gbps
- **External NICs**
  - 6x1Gbps* or 8x1Gbps**
- **IPS Accelerator**

*ASA5512-X and ASA5515-X
**ASA5525-X and higher
ASA5585-X Block Diagram

**CPU Complex**
- SSP-10: 1 CPU, 4 “cores”
- SSP-20: 1 CPU, 8 “cores”
- SSP-40: 2 CPUs, 16 “cores”
- SSP-60: 2 CPUs, 24 “cores”

**Crypto Complex**

**RAM**

**MAC 1**
- 2x10Gbps
- 4x10Gbps

**MAC 2**
- SSP-40/60
- 2x10Gbps

**Internal Switch Fabric**
- On-board 10GE interfaces*
- On-board 1GE interfaces
- Expansion Slot SSP
- 10Gbps
- 6x10Gbps

**System Bus**

**Ethernet**

*2 on SSP-10/20 and 4 on SSP-40/60
ASA5585-X Hardware Highlights

- Scalable high performance architecture
  - Flexible connectivity options with minimum restraints
  - Hash-based packet load balancing from the fabric to MAC links
  - One direction of a conn lands on same MAC link (10Gbps cap)

- Half of MAC links are dedicated to SFR/IPS/CX-SSP when inserted
  - 1x10Gbps (SSP-10/20) or 2x10Gbps links (SSP-40/60)
  - External interfaces share MAC 10GE links with on-board ports
  - Only traffic redirected to module uses dedicated ports
  - Use dedicated interface cards for port expansion
Logical ASA Firewall Diagram

- **Control Plane**: Network infrastructure, management, audit, application inspection
- **Session Manager**: Rule checks, connection creation, policy establishment
- **Fastpath**: Existing connections, policy enforcement, audit
- **Data Link**: “External” network connectivity

---

Cisco live!
Data Link Layer
Data Link Layer Overview

• “Entrance” to the firewall
  • External Ethernet ports, MAC uplinks, or backplane connection
  • 1GE/10GE have different capacities but similar behavior

• Ethernet Network Interface Controllers (NICs) on ASA
  • High level of abstraction to upper layers
  • No CPU involvement
  • First In First Out (FIFO) queues at the “wire”
  • Receive (RX) and Transmit (TX) descriptor rings point to main memory
ASA Ingress Frame Processing

- Frames are received from wire into ingress FIFO queues
  - 32/48/64KB on 1GE (except management ports), 512KB on 10GE

- NIC moves frames to main memory through RX rings
  - Each ring slot points to a main memory address ("block" or "buffer")
  - Single RX ring per 1GE (255 or 512 slots) except ASA5585
  - Multiple RX rings per 10GE (512 slots per ring) with hashed load-balancing
  - Shared RX rings on 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
ASA 1GE NIC Architecture

1. Ethernet frame arrives on the wire
2. Placed at queue tail
3. Moved from FIFO queue to memory block via RX ring
4. Pulled by CPU for processing
5. RX ring slot refilled

Buffer Blocks (fixed size)

Ingress FIFO (Kbytes)
Ingress Load-Balancing on ASA5585-X 10GE MAC

- Select Interface 0, RX Ring 0 always
- Select Interface 0, RX Ring 3 based on source/destination IP hash
- Select Interface 1, RX Ring 2 from source/destination IP and TCP/UDP port hash

Other than IPv4/IPv6

IPv4/IPv6 Other than TCP/UDP

TCP/UDP

RX Rings

0  1  2  3

10GE Interface 0 (single ingress FIFO)

RX Rings

0  1  2  3

10GE Interface 1 (single ingress FIFO)
ASA NIC Performance Considerations

- If ingress FIFO is full, frames are dropped
  - No free slots in RX ring (CPU/memory bound)
  - Unable to acquire bus (used by another component)
    - **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

- Fixed bus overhead for memory transfers
  - 30% or 80% bus efficiency for 64 or 1400 byte packets
  - Maximize frame size and minimize rate for best efficiency
Jumbo Frames

- All modern ASA models support Jumbo Ethernet frames (~9216 bytes)
  - CRC loses efficiency when approaching 12KB of data
  - Uses 9K/16KB memory blocks

```plaintext
asa(config)# mtu inside 9216
asa(config)# jumbo-frame reservation
WARNING: This command will take effect after the running-config is saved and the system has been rebooted. Command accepted.
```

- More data per frame means less overhead and much higher throughput
  - **Doubles** single-flow throughput on ASA5585-X
  - Must be implemented end-to-end for best results

- Remember TCP MSS (more on this later)
FIFO Head-of-Queue Blocking on 10GE MAC

1. One RX ring becomes “blocked”
2. Frames destined to available RX rings go through
3. Frames to a “blocked” RX ring remain in the FIFO
4. FIFO fills up, and all incoming traffic to 10GE uplink is dropped

10GE MAC operation before 8.4(5), 9.0(2), and 9.1(2) software

Upgrade ASA5585-X to mitigate FIFO head-of-queue 10GE CPU uplink blocking by head-dropping frames destined to a “blocked” RX ring

ciscoasa# show interface detail | begin Internal-Data
Interface Internal-Data0/0 ":", is up, line protocol is up
[...
  0 input errors, 0 CRC, 0 frame, 304121 overrun, 0 ignored
[...
Queue Stats:
  RX[00]: 537111 packets, 650441421 bytes, 0 overrun
    Blocks free curr/low: 511/211
  RX[01]: 47111 packets, 63364295 bytes, 0 overrun
    Blocks free curr/low: 511/478
  RX[02]: 95143 packets, 127586763 bytes, 0 overrun
    Blocks free curr/low: 511/451
  RX[03]: 101548 packets, 114139952 bytes, 304121 overrun
    Blocks free curr/low: 511/432

10GE MAC Upgrade ASA5585-X to mitigate FIFO head-of-queue 10GE CPU uplink blocking by head-dropping frames destined to a “blocked” RX ring
NIC Egress Frame Processing

- After processing, CPU places the pointer to a packet block in the next available slot on the egress interface’s TX ring
  - Same quantities as RX rings (except ASASM and 1GE on ASA5585)
  - Shared rings on MACs (ASA5585/SM) and 1GE uplink (ASA5505)
  - Software TX rings are used for Priority Queuing
    - **Underrun** drops when TX ring is full

- NIC moves frames into the egress FIFO queue
  - 16/40/48KB for 1GE and 160KB for 10GE
Key ASA Interface Statistics

asa# show interface GigabitEthernet0/2
Interface GigabitEthernet0/2 "", is up, line protocol is up
  Hardware is i82574L rev00, BW 1000 Mbps, DLY 10 usec
  Full-Duplex(Full-duplex), 1000 Mbps(1000 Mbps)
  Input flow control is unsupported, output flow control is off
  [...]  
  5993589870 packets input, 495741647452 bytes,
  Received 1463633 broadcasts, 0 runts, 0 giants
  625160 input errors, 0 CRC, 0 frame, 625160 overrun,
  0 ignored, 0 abort
  0 pause input, 0 resume input
  0 L2 decode drops
  6073287474 packets output, 800610299578 bytes,
  0 pause output, 0 resume output
  0 output errors, 0 collisions, 1 interface resets
  0 late collisions, 0 deferred
  0 input reset drops, 0 output reset drops
  input queue (blocks free curr/low): hardware (467/362)
  output queue (blocks free curr/low): hardware (503/120)

asa5585# show interface detail
Interface Internal-Data0/0 "", is up, line protocol is up
  Hardware is i82599_xaui rev01, BW 10000 Mbps, DLY 10 usec

Times unable to move ingress frame to memory (not necessarily drops)
Dropped frames due to ingress FIFO full
Dropped frames due to TX ring full
RX and TX rings
Check Internal-Data MAC interfaces for errors on ASA5585/SM
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
Packet Rates and Overruns

• High 1-minute input packet rates with a small average packet size may signal approaching oversubscription
  • Average values discount microbursts
  • Reported rate of ~20-60K of 100-250 byte packets “per second” on 1GE
  • About 8-10 times as many on 10GE

• Single interface overruns imply interface-specific oversubscription

• Overruns on all interfaces may mean several things
  • Interface oversubscription
  • CPU oversubscription on a single-core system
  • Uneven CPU load distribution on a multi-core system
  • Memory block exhaustion
Troubleshooting Interface Oversubscription

- Establish traffic baseline with a capture on the switch port
  - Connection, packet, and bit rates
  - Per application and protocol, per source and destination IP
- Cisco Network Analysis Module (NAM)
  - High performance
  - Threshold based alerts
- Block confirmed attackers on edge router
- Legitimate application may cause bursty traffic
Case Study: Bursty Traffic Analysis in Wireshark

**Problem:** Overruns are seen incrementing on the outside 1GE interface of an ASA. Both bit and packet per second rates are low.

1. Collect SPAN packet capture on the **upstream switchport** to analyze incoming traffic.

2. Open capture in Wireshark and check **packet rate graph**.

- Default packet rate measurement interval is **1 second**
- Overruns are not expected
- ~8000 packets/sec **peak** rate
- ~5000 packets/sec **average** rate

---

**Graph Details:**
- X-axis: Time (in seconds)
- Y-axis: Number of packets
- **Statistics** tab in Wireshark
  - **IO Graphs**
  - **Tick interval:** 1 sec
  - **Pixels per tick:** 10

---

BRKSEC-3021 © 2015 Cisco and/or its affiliates. All rights reserved. Cisco Public 33
Case Study: Bursty Traffic Analysis in Wireshark

Packet activity starts at ~7.78 seconds into the capture and spikes to peak shortly after.

~98 packets/ms peak rate is equivalent to 98,000 packets/sec!

3. Set packet measurement rate to 0.001 seconds (1 millisecond) to see microbursts

4. Spike of conn creation activity from a particular host followed by bursty transfers caused overruns
EtherChannel

- Up to 16 active port members per EtherChannel in **ASA 9.2(1)+**
  - Up to 8 active and 8 standby ports in **ASA 8.4(1)+**
  - Best load distribution with 2, 4, 8, or 16 port members is now a myth
  - Not supported on ASA5505

- Effective against interface-bound oversubscription
  - Distributes ingress load across multiple FIFO queues and RX rings
  - One direction of a single flow always lands on the same link
  - No effect aside from throughput aggregation on ASA5585-X
Flow Control

- IEEE 802.3x mechanism to adjust ingress data rate in Ethernet
  - Receiver uses Pause (XOFF) or Resume (XON) frame to control transmission
  - The duration of the Pause is specified in the frame
  - The frame is processed by the adjacent L2 Sender device (switch)

- ASA appliances support “send” flow control on 1GE/10GE interfaces
  - Virtually eliminates overrun errors
  - Must enable “receive” flow control on the adjacent switch port
  - Best to enable speed/duplex auto negotiation on both sides
  - Tune low/high FIFO watermarks for best performance (except 5585)
Enabling Flow Control on ASA

```bash
asa(config)# interface GigabitEthernet0/2
asa(config-if)# flowcontrol send on 10 12 26624
```

Changing flow-control parameters will reset the interface. Packets may be lost during the reset. Proceed with flow-control changes?

```bash
asa# show interface GigabitEthernet0/2
Interface GigabitEthernet0/2 "", is up, line protocol is up
   Hardware is i82574L rev00, BW 1000 Mbps, DLY 10 usec
   Auto-Duplex, Auto-Speed
   Input flow control is unsupported,
   Available but not configured via nameif
   MAC address 0001.aaaa.bbbb, MTU not set
   IP address unassigned
   36578378 packets input, 6584108040 bytes, 0 no buffer
   Received 0 broadcasts, 0 runts, 0 giants
   0 input errors, 0 CRC, 0 frame, 0 overrun,
   0 pause input, 0 resume input
   0 L2 decode drops
   4763789 packets output, 857482020 bytes, 0 underruns
   68453 pause output, 44655 resume output
   0 output errors, 0 collisions, 2 interface resets
```

Optional low FIFO watermark in KB
Optional high FIFO watermark in KB
Optional duration (refresh interval)
Flow control status
No overruns
Pause/Resume frames sent
Packet and Connection Processing
ASA Packet Processing

- CPU Data Path thread “walks” RX rings to process packets sequentially
  - No separate Control Path thread on single-core platforms
  - Packets hashed to load-balancing queues before actual processing

- Packets remain in the same allocated memory buffers (“blocks”)
  - 2048 byte blocks for ASA5505
  - 1550 byte blocks for ports on most other ASA models
  - 1550/9344 byte blocks for ASASM
  - 9344/16384 byte blocks with Jumbo frames enabled

- Other features use the memory blocks as well
  - Global block depletion or busy CPU starve RX/TX rings and cause packet drops
ASA Data Path with Multiple Cores

• Lower-end platforms may only allocate one core to ASA process

ASA5545-X# show cpu core
Core  5 sec  1 min  5 min
Core 0  0.0%  0.0%  0.0%

• Each core runs a Data Path thread to “walk” the RX rings
  • Exclusively attach to a RX ring and pull a number of packets before moving on
  • All packets from a single connection must be worked by one core at a time

• CPU Complex is underutilized if available cores exceed active RX rings
  • Adjust load-balancing to make Data Path release the RX ring after pulling one packet
  • Negative impact with a small number of connections (<64)

asa(config)# asp load-balance per-packet auto

Dynamic selection based on traffic pattern in ASA 9.3(1)+
ASA5585 Multi-Core Load Balancing

- ASA5585-X and ASASM are designed to balance number of cores and RX rings
  - Static RX rings maintained on MAC uplinks, not external interfaces
- Per-packet load-balancing may help with uneven RX ring load

```
ciscoasa# show interface detail | begin Internal-Data
Interface Internal-Data0/0 "", is up, line protocol is up
[...]
0 input errors, 0 CRC, 0 frame, 304121 overrun, 0 ignored, 0 abort
[...]
Queue Stats:
  RX[00]: 537111 packets, 650441421 bytes, 304121 overrun
    Blocks free curr/low: 511/211
  RX[01]: 47111 packets, 63364295 bytes, 0 overrun
    Blocks free curr/low: 511/478
  RX[02]: 95143 packets, 127586763 bytes, 0 overrun
    Blocks free curr/low: 511/451
  RX[03]: 101548 packets, 114139952 bytes, 0 overrun
    Blocks free curr/low: 511/432
```

Overruns are seen on MAC uplinks
RX ring 0 is utilized more than other RX rings
Legacy CPU Load Balancer Operation

1. Ingress packets hashed from RX rings into queues

2. CPU cores step through the queues and process packets

3. A queue fills up due to heavy traffic

4. System stops polling “offending” RX rings until the queue is empty

5. Packets to the blocked RX rings are dropped

32,000 queues of 1,000 elements each

CPU Complex

Core 0

Core 1

Core 2

Core 3

Ingress FIFO
Optimized CPU Load-Balancer Operation

- RX rings sending packets to a full queue are no longer blocked in ASA 9.1(4)+
  - Packets are dropped from the full load-balancer queue instead
  - Use with asp load-balance per-packet auto with single-flow oversubscription

- System forwarding performance is significantly improved under load
  - Drops from high-rate flows are localized to their corresponding load-balancer queues
  - 32,000 queues and hash outcomes distribute work very fairly
  - Chances of a “good” and an “offending” flows colliding drop from >3% to ~0.003%

- Performance testing results showed consistent improvement across the board
  - Up to 8.5% increase in TCP throughput; up to 3.5% increase in connection rate
  - Extremely effective in combating single-flow DDoS attacks and oversubscription
Control Plane in Multiple Core ASA

- Control Path process is run in turns by every core

- Data Path escalates processing requests that require specialized handling
  - To-the-box traffic (management, AAA, Failover, ARP)
  - Application Inspection
  - TCP Syslog
  - Everything else not accelerated through Data Path

```
asa# show asp multiprocessor accelerated-features
```

- Control Path should be avoided
  - Much lower throughput than Data Path
  - Unnecessary load may affect critical components (ARP, Failover)
## Multi-Core ASA Control Path Queue

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

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

- **Request queue**: The queue where requests are initially stored.
- **Requests in queue**: The number of requests currently in the queue.
- **Max requests ever in queue**: The maximum number of requests that were ever in the queue.
- **Individual event**: Each event type, such as "Punt Event" or "Identity-Traffic Event".
- **Allocation attempts**: The number of times allocation attempts were made for each event type.
- **No memory**: The number of times there was not enough memory to allocate.
- **Blocks put into queue**: The number of blocks put into the queue.
- **Times queue limit reached**: The number of times the queue limit was reached.
New and Existing Connections

• Ingress packets are checked against the connection table
  • Fastpath works with known conn parameters (like NAT)
  • Sent to Session Manager if no match

• Connection creation is the most resource consuming step
  • ASA5585 SSP-60: **350000 conns/sec vs 10M concurrent**
  • ACL Lookup
  • NAT/PAT xlate creation
  • Audit messages (Syslog/Netflow/SNMP)
  • Stateful failover information
Connection and Xlate Tables

• Maintained in main memory on ASA
  • Memory bound resources with ~1024 bytes per flow
  • 2M→10M max conns and 1.7M→8M max xlates in ASA 8.4+ (64 bit)

• Need to be “walked” periodically
  • Maintain timers and perform cleanup
  • Bigger tables → more processing overhead → less spare CPU capacity
  • Some 64-bit processing impact

• Avoid many stale connections
  • Encourage graceful termination in application design
  • Lower TCP timeouts only if necessary
Access Control Lists (ACLs)

• Fully expanded and compiled into a binary tree structure
  • Stored in main memory on ASA
  • Compilation process temporarily elevates Control Path load
  • No performance advantage with a particular order
  • Element reuse improves space utilization
  • Smaller interface ACLs use more memory but increase lookup performance

• Checked by Session Manager before conn creation
  • ACL size mostly impacts conn setup rate
  • More impact from conns denied by outbound ACLs
  • Existing connections are only impacted at peak memory usage
ACL Rules and Performance

- Push the bound to CPU with 64-bit software on ASA5585-X

<table>
<thead>
<tr>
<th></th>
<th>5505</th>
<th>5585-10</th>
<th>5585-20</th>
<th>5585-40</th>
<th>5585-60</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum recommended (&lt;8.3, 32bit)</td>
<td>25K</td>
<td>500K</td>
<td>750K</td>
<td>750K</td>
<td>750K</td>
</tr>
<tr>
<td>Maximum recommended (8.4, 64bit)</td>
<td>25K</td>
<td>500K</td>
<td>750K</td>
<td>1M</td>
<td>2M</td>
</tr>
</tbody>
</table>

- ASA5500-X and ASASM run only 64-bit software

<table>
<thead>
<tr>
<th></th>
<th>5506-X</th>
<th>5508-X</th>
<th>5512-X</th>
<th>5515-X</th>
<th>5516-X</th>
<th>5525-X</th>
<th>5545-X</th>
<th>5555-X</th>
<th>ASASM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum recommended</td>
<td>50K</td>
<td>100K</td>
<td>100K</td>
<td>200K</td>
<td>250K</td>
<td>300K</td>
<td>500K</td>
<td>500K</td>
<td>2M</td>
</tr>
</tbody>
</table>
Network Address Translation (NAT)

- 30-40% impact on maximum connection setup rate with dynamic PAT
  - Varying throughput impact depends on platform and traffic profile

- By default, dynamic PAT xlates have a 30-second idle timeout
  - Single global IP (65535 ports) allows about 2000 conn/sec for TCP and UDP

- Per-Session Xlates in **ASA 9.0(1)+** allows immediate reuse of the mapped port
  - Unless upgraded, enabled by default for all TCP and DNS connections

```
ciscoasa# show run all xlate
xlate per-session permit tcp any4 any4
xlate per-session permit tcp any4 any6
xlate per-session permit tcp any6 any4
xlate per-session permit tcp any6 any6
xlate per-session permit udp any4 any4 eq domain
xlate per-session permit udp any4 any6 eq domain
xlate per-session permit udp any6 any4 eq domain
xlate per-session permit udp any6 any6 eq domain
```
ACL and NAT Performance Optimization

• Disable Object Group expansion with extremely large ACLs to preserve memory

\[
\text{asa(config)}\# \text{object-group-search access-control}
\]
INFO: For large ACLs, this operation may take a while to complete.

• Elevated Control Path CPU load only during the conversion process
• No change in CPU impact on new connection creation rate
• Per-rule ACL hitcount is no longer available for object group members

• Lower compilation impact on new connections with Transactional Commit Model
  • Delays changes to current ACL and NAT tables until compilation is complete
  • For large ACLs in 9.1(5)+ with asp rule-engine transactional-commit access-group
  • For large NAT configuration in 9.3(1)+ with asp rule-engine transactional-commit nat
Audit Messages

• Additional CPU load from messages or packets generated by the firewall
  • Most impact from conn creation (syslog) or polling (SNMP)
  • SNMP and TCP syslogs impact Control Path on multi-core ASA
  • Less impact from Netflow than syslog on ASA

• Packets generated by firewall create load on the network
  • Netflow minimizes per-packet overhead by bundling data
  • Binary data takes up less space than ASCII strings
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

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

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)

1 connection:
32 syslog messages
26+ packets sent
100K connections/sec: 2.8Gbps
Case Study: Logging Optimization

- Not logging to buffer unless troubleshooting
- Console logging is a bottleneck (low rate)
- Using minimum number of syslog servers and Netflow collectors
- Do not duplicate syslogs and Netflow data
- Send only certain syslogs as SNMP traps
- Not all SNMP servers need to receive traps
- Reduce severity level for syslogs
- Single syslog per block of PAT ports in ASA 9.5(1)

```
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
nat (inside, outside) source dynamic any pat-pool POOL block-allocation

%ASA-6-305014: Allocated TCP block of ports for translation from inside:10.0.0.1 to outside:192.0.2.0/1024-1535
```
VPN Connection Acceleration

- IPSec crypto operations are fully hardware accelerated
- SSL VPN is accelerated in Data Path since **ASA 9.0(1)** software
  - Significant throughput gain for AnyConnect, Smart Tunnel, and Port Forwarding
  - Very minor throughput gain for Clientless SSL VPN
- Hardware Crypto Engine “bias” tuning on ASA5545/55-X, 5585-X, and ASASM
  - IPSEC/SRTP by default, SSL, or Equally Balanced
  - TLS session may drop during mode change

```plaintext
ciscoasa(config)# crypto engine accelerator-bias ssl
```
Load-Sharing: Active/Active Failover

- Share the load with active contexts on each firewall
  - Separate different networks or traffic categories
  - Avoid asymmetric routing and context cascading
  - Useful against interface induced oversubscription
  - Risk of a major performance hit after a failover event

- CPU and memory impact with stateful failover
  - CPU load from conn and xlate management
  - Memory usage due to features and conn/xlate tables
  - Keep HTTP conn replication disabled for best results
ASA Clustering

• **Up to 16** identical ASA appliances combine in one traffic processing system
  • 320Gbps of real-world TCP traffic, 96M connections, 2.5M connections per second

• Preserve the benefits of failover
  • Feature license aggregation across entire cluster
  • Virtual IP and MAC addresses for first-hop redundancy
  • Centralized configuration replicated to all members
  • Connection state preserved after a single member failure

• Implement true scalability in addition to high availability
  • Stateless load-balancing via IP Routing or Spanned Etherchannel with LACP
  • Out-of-band Cluster Control Link to compensate for external asymmetry
  • Elastic scaling of throughput and maximum concurrent connections
Flow Processing with Clustering

1. A initiates a connection to B

2a. If UDP, query Director first

2b. If static NAT or dynamic PAT, process locally; if dynamic NAT, query Master

3. If not TCP, update Director

4. B responds to A

5a. If UDP, query Director first

5b. If TCP SYN/ACK, determine Owner and redirect using CCL

6. Response returned to A

7. Update Director
Network Protocol Interaction

- Most firewalled traffic is only inspected at **network and transport layers**
  - IP reassembly
  - Stateful inspection (TCP)
  - Pseudo-stateful inspection (UDP, ICMP)
  - Non-stateful filtering (other IP protocols, such as GRE)

- Application inspection is rare and “expensive”
- Firewall features must respect transport protocols for best performance
Transport Protocols
Unified Datagram Protocol (UDP)

• Lightweight connectionless protocol
  • 12 byte header for minimal network overhead

• Best for maximum firewall throughput
  • Minimal processing required in Data Path
  • Great for real time application requiring low latency

• Practical performance implications
  • Loss is expensive (application recovery)
  • Small packets at high rates can oversubscribe ASA interfaces
Transport Control Protocol (TCP)

• Connection oriented protocol with defined states
  • Two sides establish a transport session and exchange parameters
  • Payload bytes are numbered and acknowledged upon receipt

• Stateful firewalls easily impact performance
  • Higher processing load from conn setup to termination
  • Every packet is examined to enforce correct protocol state
  • Packet loss and re-ordering reduce throughput
TCP State Bypass on ASA

- State Bypass simplifies stateful security checks to match UDP
  - ACL-based security policy for selected connections
  - Useful to reduce processing overhead on trusted flows

- Default conn timeout is not modified on ASA
  - Trusted flows with high setup/teardown rates can quickly fill up conn table
  - Set the conn timeout to 2 minutes (default on FWSM) to match UDP

```text
policy-map BYPASS_POLICY
  class TCP_BYPASSED_TRAFFIC
    set connection advanced-options tcp-state-bypass
    set connection timeout idle 0:02:00
```
TCP Maximum Segment Size

- TCP MSS option advertises maximum payload size that endpoint will accept
- ASA adjusts TCP MSS down to 1380 bytes by default
  - Reduction in throughput with no VPN or with Jumbo frames

<table>
<thead>
<tr>
<th>Outer IP</th>
<th>ESP</th>
<th>AH</th>
<th>Inner IP</th>
<th>TCP</th>
<th>TCP Payload</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 bytes</td>
<td>36 bytes</td>
<td>24 bytes</td>
<td>20 bytes</td>
<td>20 bytes</td>
<td>1380 bytes</td>
</tr>
</tbody>
</table>

- 1500 IP MTU

80 bytes wasted on non-VPN traffic

- Disable adjustment for maximum payload per TCP segment

  `asa(config)# sysopt connection tcpmss 0`
TCP Single Flow Throughput

- One way TCP throughput is limited by Window and Round Trip Time (RTT)

\[
\text{Max Single TCP Flow Throughput [bps]} = \frac{\text{TCP Window [bytes]}}{\text{RTT [seconds]}} \times 8 \text{ [bits/byte]}
\]

- Bandwidth Delay Product
Case Study: TCP Flow Throughput

TCP Window

TCP Data

TCP ACK, TCP Window

192.168.1.101

172.16.171.125

115.340 - 5.24 = 110.01ms

Matching ACK
Seq + TCP Length

Receive Window
65535 bytes

Max Single TCP Flow Throughput = \frac{65535 \text{ bytes}}{0.1101 \text{ sec}} \times 8 \text{ bits/byte} = 4.75 \text{ Mbps}
TCP Window Scaling

• TCP Window Scale (WS) option expands Window size
  • Both sides must independently advertise their Scaling Factor
  • Multiply advertised Receive Window size by 2Scaling Factor
  • Up to 32 bits total Window size (~4.3 GBytes)

Window Scaling offered with Scaling Factor of 0 (do not multiply advertised window)

Window Scaling accepted with Scaling Factor of 3 (multiply advertised window by 8)

Optimal TCP Window Size [bytes] = Minimum Link Bandwidth [bps] \times RTT [seconds]

\[
\text{Optimal TCP Window Size [bytes]} = \frac{\text{Minimum Link Bandwidth [bps]}}{8 \text{[bits/byte]}} \times \text{RTT [seconds]}
\]
TCP Selective Acknowledgement

- TCP throughput is significantly reduced by packet loss
  - All data after the lost segment must be retransmitted
  - Takes RTT to learn about a lost segment

- TCP Selective Acknowledgement (SACK) prevents unnecessary retransmissions by specifying successfully received subsequent data

```
Sequence number: 56427298
Acknowledgement number: 1069276099
Header length: 32 bytes
Flags: 0x10 (ACK)
   window size: 64512
Checksum: 0x6435 [correct]
Options: (12 bytes)
   NOP
   NOP
   SACK: 1069277089-1069287090
      left edge = 1069277089
      right edge = 1069287090
```

Retransmit data starting from this byte

Do not retransmit this later data as it has been received successfully
Application Inspection
Application Inspection Engines

- Highest level of security checks impacts performance most
- Most matching traffic is redirected to Control Path
  - HTTP, DNS, and ICMP are inspected in Data Path on multi-core ASA
  - SIP is also inspected in Data Path in 9.4(1)+ and GTP will be in 9.5(1)+
- Additional TCP Normalization of inspected traffic
  - Packets ordered within the flow
  - Per-flow buffer based on TCP MSS and Window Size on ASA
Case Study: SQL*Net Inspection

• SQL*Net inspection degrades flow and firewall performance when data is sent over Control connection

  asa# show service-policy | include sqlnet
  Inspect: sqlnet, packet 2184905025, drop 0, reset-drop 0, v6-fail-close 0
  asa# show service-policy | include sqlnet
  Inspect: sqlnet, packet 2192153131, drop 0, reset-drop 0, v6-fail-close 0

• Enable only selectively

  asa(config)# access-list SQL permit tcp any host 192.168.100.11 eq 1521
  asa(config)# class-map SQL_TRAFFIC
  asa(config-cmap)# match access-list SQL
  asa(config)# policy-map SQL_POLICY
  asa(config-pmap)# class SQL_TRAFFIC
  asa(config-pmap-c)# inspect sqlnet

Large increments in inspected packets imply that no separate data connections are used

Define a specific class to match SQL*Net control traffic to servers that actually use secondary data connections
TCP Proxy

- TCP Proxy module fully reassembles application messages before inspection
  - Used by IM, H.225, SIP, Skinny, RTSP, CTIQBE, SunRPC, and DCERPC engines

- Major performance impact due to the level of processing
  - Spoofed TCP ACK segments to get full messages
  - Segments held in a per-flow buffer (64KB on ASA)
  - Advantages of TCP WS are eliminated for the flow (<16KB window)
  - Worst impact from IM Inspection (matches all TCP ports by default)

- Limit the use of inspection engines that rely on TCP Proxy
VoIP Protocol Inspection

• Most impact during phone registration and call setup messaging
  • SIP performs better than Skinny due to less overhead
  • Further registration and call setup rate hit with TLS Proxy

• Media connections (RTP/RTCP) are handled in Data Path
  • High rate of small UDP datagrams
  • Control and associated media conns used to be handled by same core
  • No more than 2000 RTP/RTCP flows per single control connection before 9.4(1)

• Only SIP inspection is multi-core accelerated in ASA 9.4(1+)
  • ~8000 calls per second on ASA5585-X SSP-60, 20000+ with a 4-unit cluster
  • RTP/RTCP flows for a single SIP control connection are no longer constrained
FirePOWER Services Module

• Dedicated hardware module for ASA5585-X or ASA5500-X software upgrade
  • Threat centric security for NGFW/NGIPS deployments
  • Powerful AVC, URL Filtering, and Advanced Malware Protection capabilities
  • Fully scalable with ASA clustering

• Selective traffic redirection from ASA for flexible performance
  • **Still** need application inspection on ASA for NAT and secondary channels
  • TCP ordering is performed by the ASA to optimally use FirePOWER resources
  • FirePOWER has ability to selectively offload flow processing back to ASA

%ASA-5-434004: SFR requested ASA to bypass further packet redirection and process flow from inside:192.168.1.101/12987 to outside:192.0.2.100/443 locally
Cisco Cloud Web Security

- Cloud-based HTTP/HTTPS Content Filtering solution
  - Introduced in **ASA 9.0(1)** software
  - Original request redirected to a CWS tower using a destination rewrite
  - ASA supplies pre-NAT IP and user identity information
  - **Not compatible** with traffic redirected to FirePOWER services

- Significant performance advantages over legacy URL Filtering
  - Applied in Data Path on multi-core platforms
  - External cloud processing, policy configuration, and reporting
Closing Remarks
Maximizing Firewall Performance

• Avoid congestion at Data Link
• Target Fastpath
• Minimize conn creation activity
• Maximize payload size
• Optimize at Transport layer
• Selectively apply Advanced Features

Combine effective security policies with scalable network and application design to get the most from your firewall!
Q & A
Reference Slides

• These helpful materials could not be included into the session due to time constraints

• Many slides cover legacy products and features that you may still use

• Enjoy!
Pyramid of Firewall Resources

Level of Inspection

Max sessions

Bytes/sec

Packets/sec

Min latency

Desired Metrics (variable)

Firewall Resources (fixed volume)

“Fast, Good, or Cheap. Pick Two!”
ASA5510-5550 Block Diagram

**CPU**
- **Bus 1**
  - **Internal NIC**
    - 1Gbps
  - **Expansion Slot**
    - 4GE, AIP, or CSC

**Bus 0**
- **External NICs**
  - 4x1Gbps
- **On-board Interfaces**
  - 4x1GE*
- **Crypto Engine**

**System Bus**
- **Management 0/0 FE**
- **Ethernet**

*2xFE+2xGE on ASA5510 with Base license
** Fixed 4GE-SSM on ASA5550 only
CounterPath Mojo 5580 Block Diagram

CPU Complex
- 5580-20: 2 CPUs, 4 cores
- 5580-40: 4 CPUs, 8 cores

RAM

Management 2x1GE

I/O Bridge 2
- Slots 7-8

I/O Bridge 1
- Slots 3-6

Crypto Engine

System Bus
ASA5580 Hardware Highlights

- Multilane PCI Express (PCIe) slots
  - Use slots 7, 5, and 8 (x8, 16Gbps) for 10GE cards first
  - Use slots 3, 4, and 6 (x4, 8Gbps) for 1GE/10GE cards

- Ensure equal traffic distribution between the I/O bridges
  - With only two active 10GE interfaces, use slots 7 and 5

- Keep flows on same I/O bridge with 3+ active 10GE ports
  - Place interface pairs on the same card
Simplified FWSM Block Diagram

- Control Point
- Network Processor 1
- Network Processor 2
- Network Processor 3
- Rule Memory
- Switch Backplane
- 6x1GE Etherchannel
- System Bus
- Ethernet

Connections:
- 2x1Gbps
- 1Gbps
- 4Gbps
- 3x1Gbps
- 3x1Gbps
- 3x1Gbps
- 1Gbps
- 1Gbps
ASA5510-5550 Hardware Highlights

• With a 4GE-SSM, 1Gbps link is shared between 4x1GE ports
  • No throughput issue on ASA5510-5540
  • On ASA5550, get 1.2Gbps between a 4GE-SSM port and an on-board interface
  • On-board interfaces are better for handling high packet rates

• Content Security Card (CSC) may starve other traffic
  • File transfers proxied over a dedicated 1GE connection
ASA 5500-X Hardware Highlights

- Direct Firewall/IPS integration for higher performance
  - Future application expansion

- Switched PCI connectivity to all interfaces

- Management port is only for management
  - Shared between Firewall and IPS
  - Very low performance
FWSM Hardware Highlights

- Distributed Network Processor complex
  - Fastpath (NP 1 and 2), Session Manager (NP 3), Control Point

- Etherchannel connection to the switch backplane
  - An external device with 6x1GE ports for all intents and purposes

- No local packet replication engine for multicast, GRE, …
  - SPAN Reflector allows Sup to replicate egress packets
  - Over 3 FWSMs in a chassis may cap throughput under full load
ASA Services Module Hardware Highlights

- Architecture similar to ASA5585-X with SSP-20
  - Hash-based load balancing to MAC links with 10Gbps unidirectional flow cap
  - Minor throughput impact due to extra headers (VLAN/internal)
  - Data link subsystem optimized for extra cores

- Improved switch integration over FWSM
  - No switch-side Etherchannel
  - Local egress packet replication
Frame Processing on FWSM

- Switch-side load-balancing across a 6x1GE Etherchannel
  - Check packet counters on the member ports to gauge load
  - Tweak the global load-balancing algorithm if necessary

- Incoming frames are sent to ingress queues on NP 1 and 2
  - Send Flow Control is always enabled
  - NPs send Pause frames on all GE ports (3 each) when congested

- Jumbo frames (up to 8500 bytes) give best performance
  - Set the logical interface MTU, no other commands required
  - Respective PortChannel interface will still show MTU of 1500
FWSM Backplane Etherchannel

```
switch# show firewall module 1 traffic
Firewall module 1:

Specified interface is up line protocol is up (connected)
  Hardware is EtherChannel, address is 0012.7777.7777 (bia 0012.7777.7777
  MTU 1500 bytes, BW 6000000 Kbit, DLY 10 usec,
    reliability 255/255, txload 1/255, rxload 1/255
Encapsulation ARPA, loopback not set
Full-duplex, 1000Mb/s, media type is unknown
input flow-control is on, output flow-control is on
Members in this channel: Gi1/1 Gi1/2 Gi1/3 Gi1/4 Gi1/5 Gi1/6
Last input never, output never, output hang never
Last clearing of "show interface" counters never
Input queue: 0/2000/0/0 (size/max/drops/flushes); Total output drops: 0
Queueing strategy: fifo
Output queue: 0/40 (size/max)

5 minute input rate 2000 bits/sec, 2 packets/sec
5 minute output rate 6000 bits/sec, 9 packets/sec

Send Flow Control is enabled
Member ports <FWSM slot>/[1-6]
Input: from the FWSM
Output: to the FWSM
```

25288362 packets input, 3304220283 bytes, 0 no buffer
Received 10449 broadcasts, 0 runts, 0 giants, 0 throttles

[...]

Cisco Live!
Case Study: Collisions on ASA

- A Full duplex interface should never see collision errors
  - Collision errors on a Full duplex interface imply that the other side is running at 100Mbps and in Half duplex
  - Sudden drop in throughput after unknown uplink changes

- Speed can be sensed passively, but duplex cannot
  - If the remote side is set to 100Mbps, it will not transmit any negotiation information
  - If the local port is set to auto negotiate, it will sense 100Mbps but use Half duplex

- Auto negotiation is recommended on all interfaces
  - Hard code only if the remote side is hardcoded (i.e. 100Mbps/Full)
QoS on ASA

- Police to limit the throughput of certain traffic to “reserve” bandwidth for other important traffic
  - Applied in CPU (after packet is permitted on input and before NIC on output)
  - Not effective against overrun and underrun errors

- Strict priority queuing may starve best effort traffic
  - Not supported on 10GE interfaces on ASA5580
  - Affects all interfaces on ASA5505
  - Very limited benefit for Internet traffic

- Shape outbound bandwidth for all traffic on an interface
  - Useful with limited uplink bandwidth (i.e. 1GE link to 10Mb modem)
  - Not supported on high-performance ASA558x models
Case Study: Downstream QoS

- QoS on downstream switch can be used as a reactive measure against ASA interface oversubscription
  - Police output rate to less than the maximum forwarding capacity
  - Limit output burst size to prevent input FIFO overflow

\[
\text{Burst [bytes]} = \frac{\text{Rate [bps]}}{8 \text{ [bits/byte]}} \times \text{Token Refill Frequency [sec]}
\]

- FIFO size is sufficient for the maximum link burst size
  - Assume a 1GE interface with 32 KBytes of input FIFO
  - Assume a Cisco switch with 0.25ms burst token refill frequency

\[
\text{Burst} = \frac{1 \text{Gbps}}{8 \text{ bits/byte}} \times 0.00025 \text{ sec} = 32 \text{ KBytes}
\]

- Limiting burst size relieves FIFO load but reduces throughput
## Memory Blocks on ASA

### show blocks

<table>
<thead>
<tr>
<th>SIZE</th>
<th>MAX</th>
<th>LOW</th>
<th>CNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>700</td>
<td>699</td>
<td>700</td>
</tr>
<tr>
<td>4</td>
<td>300</td>
<td>299</td>
<td>299</td>
</tr>
<tr>
<td>80</td>
<td>919</td>
<td>908</td>
<td>919</td>
</tr>
<tr>
<td>256</td>
<td>2100</td>
<td>2087</td>
<td>2094</td>
</tr>
<tr>
<td>1550</td>
<td>9886</td>
<td>411</td>
<td>7541</td>
</tr>
<tr>
<td>2048</td>
<td>3100</td>
<td>3100</td>
<td>3100</td>
</tr>
<tr>
<td>2560</td>
<td>2052</td>
<td>2052</td>
<td>2052</td>
</tr>
<tr>
<td>4096</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>8192</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>16384</td>
<td>152</td>
<td>152</td>
<td>152</td>
</tr>
<tr>
<td>65536</td>
<td>16</td>
<td>16</td>
<td>16</td>
</tr>
</tbody>
</table>

Currently allocated blocks ready for use

1550 byte blocks were close to exhaustion

### show blocks interface

<table>
<thead>
<tr>
<th>Memory Pool</th>
<th>SIZE</th>
<th>LIMIT/MAX</th>
<th>LOW</th>
<th>CNT</th>
<th>GLB:HELD</th>
<th>GLB:TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>DMA</td>
<td>2048</td>
<td>512</td>
<td>257</td>
<td>257</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>DMA</td>
<td>1550</td>
<td>154</td>
<td>1540</td>
<td>1540</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Block size for RX/TX rings

Block count for RX/TX rings

Block count “borrowed” from global pool

Total blocks ever “borrowed” from global
FWSM Packet Processing

- NP 1 and 2 process packets from the input queues first
  - 32K ingress and 512K egress buffers (blocks) per NP
  - When an NP is busy processing packets, the input queue grows → drops
  - Existing connections are handled here (“Fastpath”)

- Some packets are sent up to NP3 (“Session Manager”)
  - Same kind of input queues as NP1 and 2
  - Significantly slower than NP1 and 2 due to additional code

- Control Point is a general purpose “visible” CPU on FWSM
  - Management, inspection, logging, NP control tasks, IPv6 traffic
  - Slow (300-500Mbps) compared to NP1 and 2 (>2Gbps each)
Queues and Back Pressure on FWSM

<table>
<thead>
<tr>
<th>NP</th>
<th>MAX</th>
<th>FREE</th>
<th>THRESH_0</th>
<th>THRESH_1</th>
<th>THRESH_2</th>
</tr>
</thead>
<tbody>
<tr>
<td>NP1  (ingress)</td>
<td>32768</td>
<td>32368</td>
<td>3067</td>
<td>420726</td>
<td>634224</td>
</tr>
<tr>
<td>(egress)</td>
<td>521206</td>
<td>521204</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>NP2  (ingress)</td>
<td>32768</td>
<td>32400</td>
<td>8395</td>
<td>1065414</td>
<td>758580</td>
</tr>
<tr>
<td>(egress)</td>
<td>521206</td>
<td>521183</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>NP3  (ingress)</td>
<td>32768</td>
<td>32768</td>
<td>1475</td>
<td>239663</td>
<td>2275171</td>
</tr>
<tr>
<td>(egress)</td>
<td>521206</td>
<td>521206</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Current free blocks:

- <48 free blocks seen (drop control frames)
- <80 free blocks seen (drop data frames)
- <160 free blocks seen (send Pause frames)

All 1GE interfaces on the NP send Pause frames.

```
fwsm# show np 1 stats | include pause
PF_MNG: pause frames sent (x3) : 241148
```
**Case Study: FWSM Load Distribution**

```
switch# show interfaces port-channel 305 counters etherchannel
```

<table>
<thead>
<tr>
<th>Port</th>
<th>InOctets</th>
<th>InUcastPkts</th>
<th>InMcastPkts</th>
<th>InBcastPkts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Po305</td>
<td>3950828072</td>
<td>30564771</td>
<td>347</td>
<td>12674</td>
</tr>
<tr>
<td>Gi1/1</td>
<td>44715343</td>
<td>150658</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Gi1/2</td>
<td>11967356</td>
<td>36130</td>
<td>0</td>
<td>547</td>
</tr>
<tr>
<td>Gi1/3</td>
<td>362138676</td>
<td>4308332</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Gi1/4</td>
<td>34954036</td>
<td>139910</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Gi1/5</td>
<td>12127366</td>
<td>37060</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Gi1/6</td>
<td>753640037</td>
<td>5504228</td>
<td>0</td>
<td>261</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Port</th>
<th>OutOctets</th>
<th>OutUcastPkts</th>
<th>OutMcastPkts</th>
<th>OutBcastPkts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Po305</td>
<td>9110614906</td>
<td>28806497</td>
<td>55508294</td>
<td>15214267</td>
</tr>
<tr>
<td>Gi1/1</td>
<td>1862243517</td>
<td>160979</td>
<td>19786112</td>
<td>3749752</td>
</tr>
<tr>
<td>Gi1/2</td>
<td>44080767</td>
<td>297474</td>
<td>7317</td>
<td>9678</td>
</tr>
<tr>
<td>Gi1/3</td>
<td>25638593</td>
<td>71405</td>
<td>88</td>
<td>18576</td>
</tr>
<tr>
<td>Gi1/4</td>
<td>1077459621</td>
<td>9170603</td>
<td>722861</td>
<td>7537</td>
</tr>
<tr>
<td>Gi1/5</td>
<td>25301928</td>
<td>67036</td>
<td>178</td>
<td>119849</td>
</tr>
<tr>
<td>Gi1/6</td>
<td>22258019</td>
<td>71230</td>
<td>10406</td>
<td>13608</td>
</tr>
</tbody>
</table>

Switch configuration:
```
switch# show etherchannel load-balance
EtherChannel Load-Balancing Configuration:
  src-dst-ip
  mpls label-ip
```
ASA5580 Multi-Core Load Balancing

**show cpu core**

<table>
<thead>
<tr>
<th>Core</th>
<th>5 sec</th>
<th>1 min</th>
<th>5 min</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core 0</td>
<td>18.1%</td>
<td>18.5%</td>
<td>18.7%</td>
</tr>
<tr>
<td>Core 1</td>
<td>56.8%</td>
<td>57.2%</td>
<td>56.1%</td>
</tr>
<tr>
<td>Core 2</td>
<td>5.4%</td>
<td>6.2%</td>
<td>7.4%</td>
</tr>
<tr>
<td>Core 3</td>
<td>60.7%</td>
<td>61.3%</td>
<td>63.2%</td>
</tr>
<tr>
<td>Core 4</td>
<td>1.2%</td>
<td>1.5%</td>
<td>1.4%</td>
</tr>
<tr>
<td>Core 5</td>
<td>4.1%</td>
<td>4.3%</td>
<td>4.7%</td>
</tr>
<tr>
<td>Core 6</td>
<td>25.1%</td>
<td>24.9%</td>
<td>26.1%</td>
</tr>
<tr>
<td>Core 7</td>
<td>19.0%</td>
<td>18.7%</td>
<td>20%</td>
</tr>
</tbody>
</table>

Uneven load on the 8 cores

**show nameif**

<table>
<thead>
<tr>
<th>Interface</th>
<th>Name</th>
<th>Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>Management0/0</td>
<td>management</td>
<td>100</td>
</tr>
<tr>
<td>GigabitEthernet3/0</td>
<td>outside</td>
<td>0</td>
</tr>
<tr>
<td>GigabitEthernet3/1</td>
<td>DMZ</td>
<td>50</td>
</tr>
<tr>
<td>TenGigabitEthernet5/0</td>
<td>inside</td>
<td>100</td>
</tr>
</tbody>
</table>

Only 3 data interfaces (6 RX rings)

**show conn count**

12090 in use, 30129 most used

Sufficient number of connections

**asp load-balance per-packet**
Packet Processing

• Once received from network, packets go through security policy checks
  • All processing is done by general purpose CPU(s) on ASA
  • Specialized Network Processors and a general purpose Control Point on FWSM

• Packets reside in main memory (ASA) or NP buffers (FWSM)

• An overloaded packet processing subsystem puts back pressure on the network level (Data Link)
  • Very common performance bottleneck
Logical Packet Flow Diagram

Data Path:
- **New conn?**
  - yes
    - ACL checks
    - Create Xlate
    - Create Conn
    - TCP Norm
    - Fail over
    - Audit info
  - Mgmt
- no
  - Policy checks
  - Apply NAT
  - L2/L3 lookup
  - Fastpath
  - Session Manager

Control Path:
- TCP Proxy
- App inspect
- Dyn routing
- ARP resolve
- Mgmt
- Control Path
FWSM Control Point Interface

fwsm# show nic
interface gb-ethernet0 is up, line protocol is up
  Hardware is i82543 rev02 gigabit ethernet, address is 0011.bb87.ac00
  PCI details are - Bus:0, Dev:0, Func:0
  MTU 16000 bytes, BW 1 Gbit full duplex
  255065 packets input, 8319485653316352 bytes,
    Received 0 broadcasts, 0 runts, 0 giants
    0 input errors, 0 CRC, 0 frame, 0 overrun
    0 ignored, 0 abort
  8936682 packets output, 4124492648088076288 bytes, 0 underruns
  input queue (curr/max blocks): hardware (0/7) software (0/0)
  output queue (curr/max blocks): hardware (0/20) software (0/0)

[...]

fwsm# show block
[...]

Additional Block pools for 16384 size blocks
IP Stack  1024  1023  1024
ARP Stack  512  505  512
Slow Path  5500  5495  5500
NP-CP     1024   1012  1024
Others    132   132   132

Feature block pool
Low watermark
Current availability
Signs of CP oversubscription

0 no buffer
0 overrun

FWSM Control Plane

• Control Point is a general purpose CPU on FWSM
  • Performs management, inspection, logging, and NP control tasks
  • IPv6 packets are handled here as well
  • Packets have to go through NP 3 first
  • Slow (300-500Mbps) compared to NP1 and 2 (>2Gbps each)
  • Uses 16KByte main memory blocks for all tasks

• Control Point is the “visible” CPU
  • CLI/ASDM/SNMP “CPU load”
  • Hardware NPs are insulated from general CP oversubscription but not from some critical features (ARP, Failover)
ACL Rules and Performance

- Recommended maximum to limit conn setup rate impact (<10%)
  - Up to 25% throughput impact beyond maximum recommended size
  - Throughput impact depends on conn lifetime

- Memory bound on lower-end ASA (32-bit)

<table>
<thead>
<tr>
<th></th>
<th>5505</th>
<th>5510</th>
<th>5520</th>
<th>5540</th>
<th>5550</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum recommended</td>
<td>25K</td>
<td>80K</td>
<td>200K</td>
<td>375K</td>
<td>550K</td>
</tr>
<tr>
<td>Maximum</td>
<td>25K</td>
<td>80K</td>
<td>300K</td>
<td>700K</td>
<td>700K</td>
</tr>
</tbody>
</table>
Network Address Translation

- Identity or Static NAT is best for higher performance
- Dynamic PAT and NAT mostly affect conn setup rate
  - Smaller overhead for established sessions with NAT
  - More impact from PAT on FWSM than ASA
  - Possible indirect impact from logging
- FWSM creates identity xlates by default
  - Use Xlate Bypass to better utilize limited xlate space
    
    ```
    fwsm(config)# xlate-bypass
    ```
  - Identity xlates may be needed for packet classification or inspection
ASA Crypto Operations

• Most impact during tunnel establishment
  • RSA key generation is always done in software
  • Routine IPSEC/SSL operations are hardware accelerated

• ASA5510-5550 do DH Groups 5/14/24 and 2048 bit RSA in software by default
  • Accelerated in hardware on ASA5500-X, 558x, and ASASM
  • Exponential decrease in scalability with long keys due to complexity

  asa(config)# crypto engine large-mod-accel

• Higher performance impact from SSL VPN as compared to IPSEC
  • Very heavy CPU load from Application Proxy Engine
  • ~128KB vs ~18KB of memory usage per connection
Address Resolution Protocol

• ARP is processed in Control Path on ASA
  • Data Path requests ARP resolution from Control Path while buffering original packet
  • Possible performance hit with frequent ARP calls

• ARP resolution is done by Control Point on FWSM
  • NP 1/2 request resolution without buffering original packet
  • Easy NP3 and CP oversubscription with non-existing hosts
  • Optionally create conn entries for ARP misses on UDP traffic

```
fwsm# show np all stats | include ARP Lookup
PKT_CNT: UDP ARP Lookup miss                       : 2311
PKT_CNT: ARP Lookup miss                           : 28
PKT_CNT: UDP ARP Lookup miss                       : 4781
PKT_CNT: ARP Lookup miss                           : 36
fwsm(config)# sysopt connection udp create-arp-unresolved-conn
```
Multicast

- IGMP and PIM are processed in the Control Plane
  - Use static IGMP joins where applicable for less overhead
  - ASA must not be RP and DR for both sender and receiver

- Established multicast data conns are handled in Fastpath
  - Best to “prime” a multicast flow with minimal traffic first
  - Bigger hit with small packets compared to unicast on ASA
  - Number of groups scales well with large packets
  - Number of egress interfaces directly affects performance
Inline Packet Capture

• Negligible performance impact on single-core ASAs

• Significant CPU impact with a lot of matching traffic on multi-core ASAs
  • Packets are read and displayed in Control Path
  • The necessary lock structure starves Data Path

• Several caveats on the FWSM
  • Capture ACL is always required to protect Control Point
  • Matching traffic may get re-ordered on the wire
Advanced Features

• Threat Detection statistics should only be gathered when troubleshooting specific attacks due to memory impact

• Optimize dynamic routing protocols behavior
  • Memory impact from the number of routes
  • Control Plane processing impact from updates
  • Summaries routes and minimize reconvergence

• Avoid enabling features unless necessary
  • Memory and CPU impact from one feature affects forwarding capacity of entire system
Failover

- Control traffic is handled in Control Path
  - Stateful updates are offloaded to Data Path in multi-core ASAs

- Failover relies on memory blocks, CPU, and NIC infrastructure
  - Block exhaustion may cause overruns and failovers

- Stateful Failover comes with a performance impact
  - Up to 30% reduction in max conn setup rate
  - HTTP conn replication is disabled by default (short lived)
  - Dedicated 1GE link is sufficient for up to ~300K conn/sec
  - Link latency under 10ms to avoid retransmissions
ASA Load-Sharing: External Routing

- Load-share between Routed ASAs using IP routing
  - Equal Cost Multi Path (ECMP) with dynamic routing
  - NAT/PAT with Policy Based Routing (PBR)
  - Linear performance scaling with hardware PBR and right traffic

- Some fault tolerance with dynamic routing or PBR/SLA
  - Active/Standby Failover for each member
  - Traffic loss when removing “bundle” members

- Centralized management is still a challenge
  - Use CSM Shared Policies
ASA Load-Sharing: External EtherChannel

• “Bundle” Transparent ASAs via a **through** EtherChannel
  • Source or Destination IP hashing based on direction
  • Unidirectional NAT is possible
  • Linear performance scaling when traffic balance is right

• Poor fault tolerance and management
  • LACP/PAgP for dynamic bundling
  • No Failover due to interface bring-up order for hashing
  • Requires out-of-band management

• Only works well between routers due to MAC learning
  • Static MAC mappings are required on ASA
Clustering Performance Guidelines

- Traffic symmetry through a cluster member delivers best performance
  - Match Etherchannel load-balance methods between switches and cluster
  - Avoid NAT when possible
  - Maximum conn counts and setup rates always scale worse than throughput

- Size and protect Cluster Control Link (CCL) appropriately
  - Bandwidth should match maximum forwarding capacity of each member
  - Set MTU 100 bytes above largest data interface MTU to avoid fragmentation
  - Use interface expansion cards for extra 10GE ports in ASA 9.1(2+)
  - Spanned Etherchannel interface mode offers quicker convergence
Case Study: TCP SACK and FWSM

• FWSM hides TCP sequence numbers of “inside” hosts by default
  • Feature is called TCP Sequence Number Randomization
  • Fixed offset set during conn creation and applied by Fastpath

```plaintext
fwsm(config)# policy-map global_policy
fwsm(config-pmap)# class RNDM_EXEMPT
Fwsm(config-pmap-c)# set connection random-sequence-number disable
```
TCP Windowing

• TCP Receive Window specifies the amount of data that the remote side can send before an explicit acknowledgement
  • 16 bit field allows for up to 65535 bytes of unacknowledged data

• Send and Receive Windows are managed separately in a TCP connection
  • Each side keeps its own Receive Window and updates the peer in every segment
  • Each side sets Send Window based on the last Window update from the peer and amount of data transmitted since
  • Send Window size is decremented with every data byte transmitted
  • Concept of Sliding Window allows a continuous stream of data
Firewalls and TCP Options

• Network applications should use TCP WS and SACK
  • WS enabled by default on MS Windows Vista, 7, and 2008 Server

• Firewalls should not clear TCP WS and SACK options
  • Default behavior on both ASA and FWSM
  • Check for TCP maps that may clear WS and SACK on ASA

asa# show run tcp-map
tcp-map OPTIONS_CLEAR
tcp-options selective-ack clear
tcp-options window-scale clear

fwsm# show run sysopt
[..]
sysopt connection tcp
sysopt connection tcp

WS and SACK cleared on ASA (suboptimal configuration)

WS and SACK permitted on FWSM (optimal configuration)
TCP Packet Reordering

• Out-of-order TCP segments reduce performance
  • Re-assembly effort by transit devices and receiver
  • May trigger retransmission requests

• Transit multi-path load balancing may impact order

• FWSM parallel processing architecture impacts order
  • Smaller packet of a connection may get sent ahead
  • Significant reduction in performance of TCP flows
  • Enable Completion Unit globally to **preserve** order of **most** Fastpath traffic

```
fwsrm(config)# sysopt np completion-unit
```
FWSM Completion Unit

• Completion Unit is an internal FWSM module that maintains same packet order at ingress and egress
  • Tags the frames to eliminate FWSM-induced reordering
  • Will not correct the original order of ingress traffic
  • Only works with pure Fastpath traffic
  • Will not help multicast, fragmented, or captured packets
  • Minor performance implications in corner cases

• Enable globally to maximize TCP performance

```
fwm(config)# sysopt np completion-unit
```
Case Study: ASA TCP Reordering

- Drops from reordering buffer decrease performance when the dynamic size determination is not accurate

```plaintext
asa# show asp drop | include buffer
TCP Out-of-Order packet buffer full (tcp-buffer-full) 4465608
TCP Out-of-Order packet buffer timeout (tcp-buffer-timeout) 406008

asa(config)# tcp-map ORDER_QUEUE
asa(config-tcp-map)# queue-limit 100 timeout 5
asa(config)# policy-map global_policy
asa(config-pmap)# class INCREASE_QUEUE
asa(config-pmap-c)# set connection advanced-options ORDER_QUEUE
```

- Set the buffer size statically (avoid high limits)
- Increase the timeout if needed (avoid long reordering timeouts)
- Define a very specific class (all matching flows will be ordered)

- No more space in the reordering buffer
- Segments sat in the reordering buffer too long
URL Filtering

• Performance impact due to complexity
  • Reliance on external server
  • Applied in Control Plane
  • Entire flow is ordered by TCP Normalizer
  • Complex parsing and buffering mechanisms

• Ensure that only untrusted HTTP traffic is matched

asa(config)# filter url except 192.168.0.0 255.255.0.0 172.16.0.0 255.255.0.0
asa(config)# filter url http 192.168.1.0 255.255.255.0 0.0.0.0 0.0.0.0

Exempt traffic to trusted internal servers
Only match clear text HTTP ports
URL Filtering Operation

1. HTTP GET request sent from client to WWW server

2. URL is parsed out and a request is sent to URL server

3. WWW server sends the page but ASA is waiting on URL server

4. URL server sends permit or deny

5. Actual or deny page is forwarded to client

Internet

1. HTTP GET request is forwarded outside

2. HTTP GET request is forwarded outside

3. WWW server sends the page but ASA is waiting on URL server
Case Study: URL Filtering Performance

- Limit latency and impact to URL server from firewall side

```
asa(config)# url-block block 128
```
Enable buffering of HTTP responses to reduce retransmissions (up to 128 packets)

```
asa(config)# url-server (dmz) host 172.16.1.1 protocol UDP
```
Switch to UDP to reduce load on ASA and speed up request generation rate (may overload URL server)

```
asa(config)# url-server (dmz) host 172.16.1.1 protocol TCP connections 25
```
Allow long URLs (up to 4KB) and avoid truncation that may cause a reverse DNS lookup on URL server

```
asa(config)# url-block url-size 4
asa(config)# url-block url-mempool 5000
```
Increase concurrent TCP connection count to parallelize requests (high values will impact URL server)

```
asa(config)# url-block url-size 4
asa(config)# url-block url-mempool 5000
```
Allocate memory for buffering long URLs (up to 10240KB)
Case Study: URL Filtering Performance

• Detect URL server oversubscription

asa# show url-block block statistics
[...]
Packets dropped due to
  exceeding url-block buffer limit: 26995
  HTTP server retransmission: 9950

asa# show url-server statistics | include LOOKUP_REQUEST
LOOKUP_REQUEST
  323128258
  322888813

Syslogs indicating pending URL requests

%ASA-3-304005: URL Server 172.16.1.1 request pending URL http://cisco.com

Buffered responses dropped at a high rate

Significant disparity between sent and responded URL requests
Legacy ASA Security Service Modules

- Usual IPS performance caveats for AIP-SSM/IPS-SSP
  - TCP ordering is enabled on traffic sent to IPS
  - Least impact on firewall throughput in promiscuous mode

- Content Security Card proxies transit connections
  - TCP ordering is not performed by the ASA
  - Redirect only untrusted traffic over supported TCP ports
  - Local QoS is not effective to limit proxied transfers
  - Set limits on maximum scannable file sizes for best performance
ASA Next-Generation Firewall Services (CX)

- Dedicated Application security (ASA5585-X SSP or ASA5500-X package)
  - Rich micro-application support
  - Real-time protection through Cisco SIO
  - Granular and flexible policy model

- Significant performance advantages over pattern matching on ASA
  - Up to 13Gbps AVC/WSE multiprotocol throughput with CX SSP-60
  - Scales well with applications that use non-standard ports
  - TCP ordering is not performed on ASA
  - **Still** need application inspection on ASA for NAT and secondary channels
Participate in the “My Favorite Speaker” Contest
Promote Your Favorite Speaker and You Could Be a Winner

• Promote your favorite speaker through Twitter and you could win $200 of Cisco Press products (@CiscoPress)

• Send a tweet and include
  • Your favorite speaker’s Twitter handle CiscoDCSecurity
  • Two hashtags: #CLUS #MyFavoriteSpeaker

• You can submit an entry for more than one of your “favorite” speakers

• Don’t forget to follow @CiscoLive and @CiscoPress

• View the official rules at http://bit.ly/CLUSwin
Complete Your Online Session Evaluation

- Give us your feedback to be entered into a Daily Survey Drawing. A daily winner will receive a $750 Amazon gift card.

- Complete your session surveys though the Cisco Live mobile app or your computer on Cisco Live Connect.

Don’t forget: Cisco Live sessions will be available for viewing on-demand after the event at CiscoLive.com/Online
Continue Your Education

- Demos in the Cisco campus
- Walk-in Self-Paced Labs
- Table Topics
- Meet the Engineer 1:1 meetings
- Related sessions
Thank you
TOMORROW starts here.