

# Troubleshooting Firepower Threat Defense like a TAC Engineer

Foster Lipkey, Principal Engineer, CX  
Alejandra Paez, Technical Leader, CX  
John Groetzinger, Technical Leader, CX

# Cisco Webex App

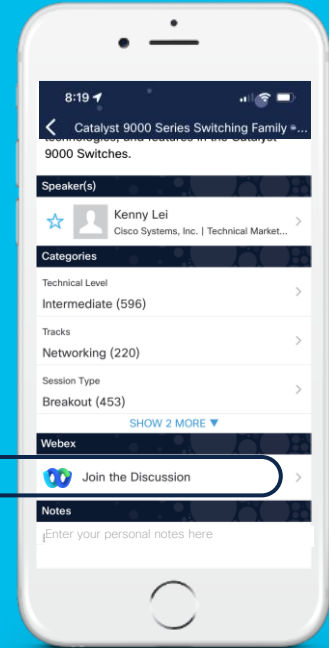
## Questions?

Use Cisco Webex App to chat with the speaker after the session

## How

- 1 Find this session in the Cisco Live Mobile App
- 2 Click “Join the Discussion”
- 3 Install the Webex App or go directly to the Webex space
- 4 Enter messages/questions in the Webex space

Webex spaces will be moderated until February 24, 2023.



# Why is FTD troubleshooting so important?

- ASA and Firepower technologies have merged into a unified solution: FTD
- FTD is more complex to troubleshoot; an understanding of both ASA and Firepower technologies is needed.
- Without expertise, there is more risk of network downtime or security breaches. Both are frustrating and impact the business.

# Presentation Objectives and Outcomes

- To combat this, today we're going to arm you with knowledge, skills, and tools to more effectively troubleshoot and resolve incidents on the Cisco FTD platform
- We encourage you to think about past or potential future experiences where you can apply these skills

Goal: Fewer late night  
troubleshooting calls





# Agenda

- Introduction
- Architecture Overview
- Path of the Packet
- Troubleshooting Tools
- Interactive Troubleshooting
- Q&A

# Your Presenters

## Foster Lipkey

TAC Security - Principal Engineer  
10+ Years of Security Experience  
Snort Expert  
Sourcefire Veteran  
Automation Enthusiast



**SOURCE**fire®



# Your Presenters

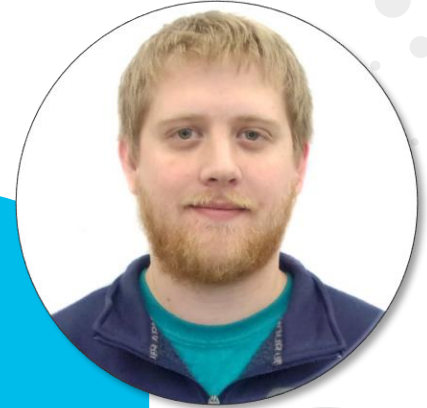
## John Groetzinger

Technical Leader for Firepower TAC

10+ Years experience in Network Security

Original Sourcefire employee

Open Source, devops and Linux enthusiast



# Your Presenters



## Alejandra Paez

Venezuela / Mexico.

Technical Leader CX Security.

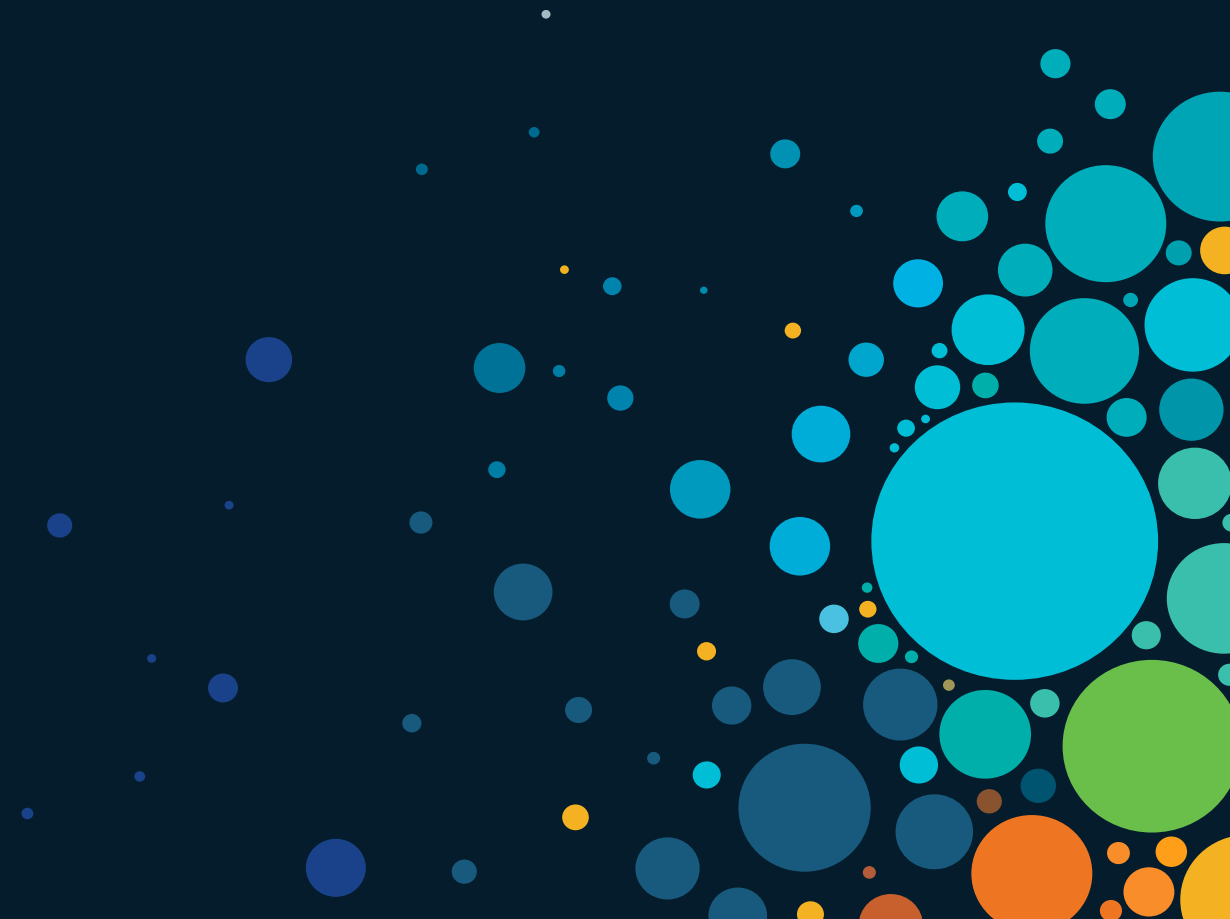
7+ years in Cisco Firepower TAC.

Passionate about Firepower NGFW Security appliances.





# Introduction



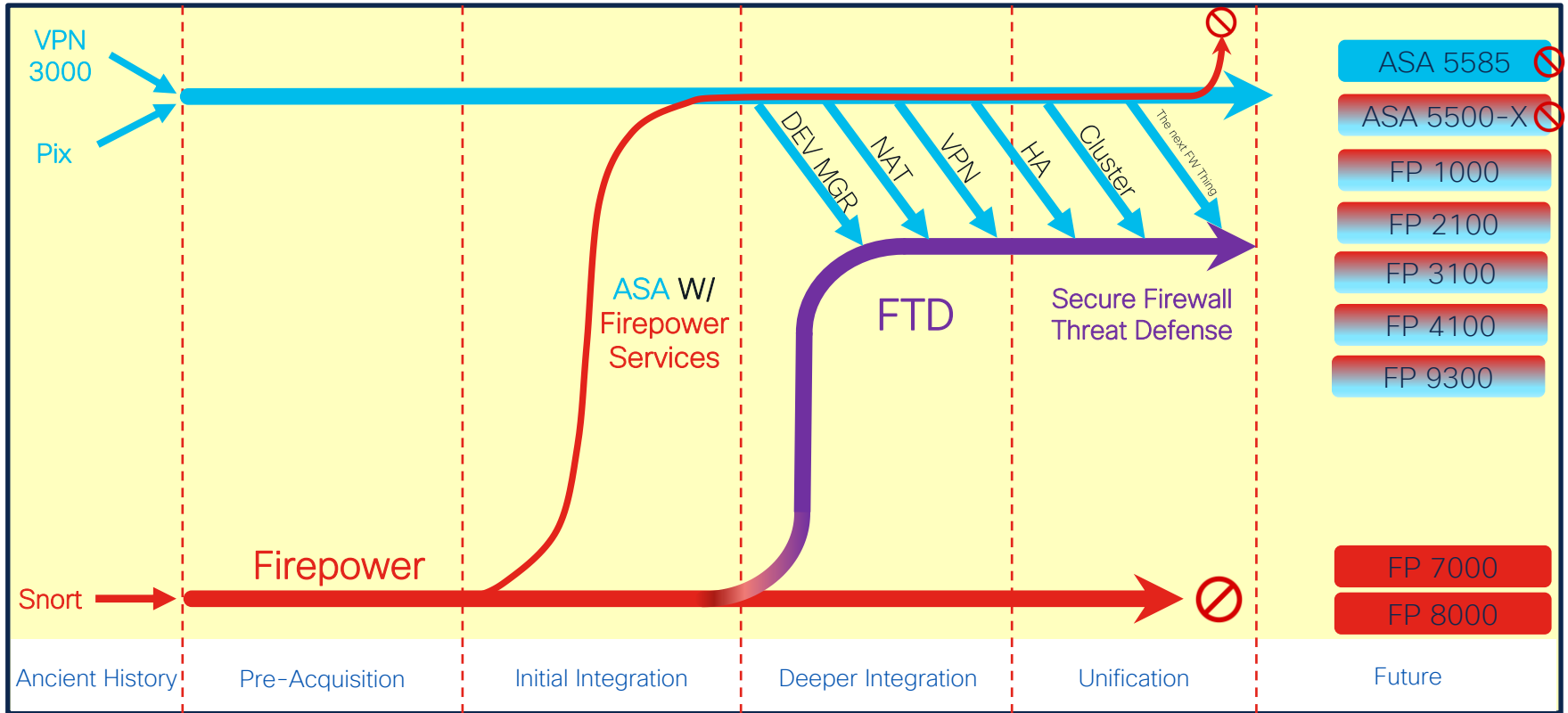
# Introduction - Presentation Focus Areas

- This is not an introductory session! General familiarity with either ASA or Firepower is assumed.
- Other Cisco Live presentations cover FTD features, design, deployment, and configuration. We are focused on product functionality and troubleshooting.
- Configuration and troubleshooting of the FXOS platform is out of scope although it will be referenced as needed.

# Introduction – Key Terminology

These terms are within the context of Firepower Threat Defense.

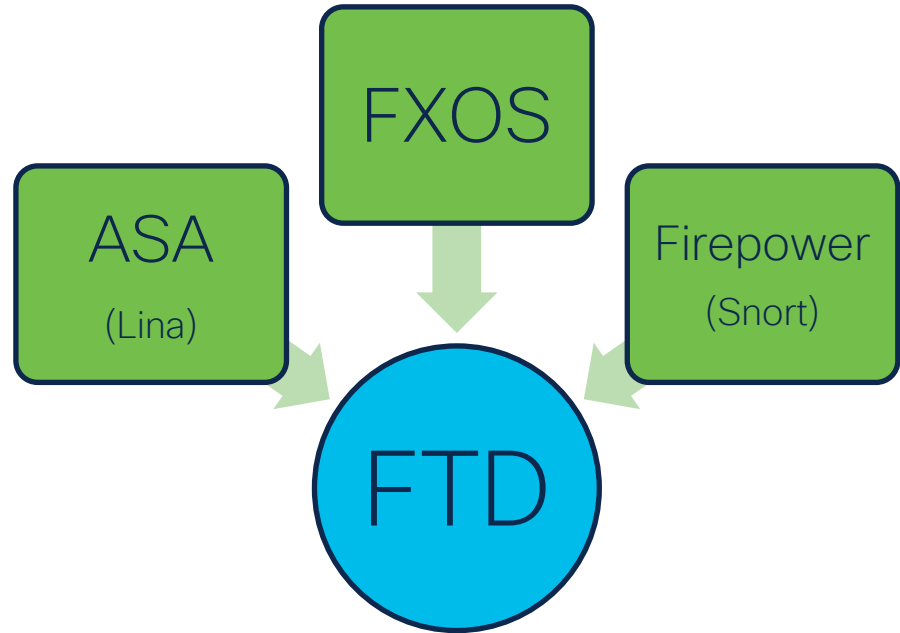
Term	Definition
Lina	Underlying ASA-derived process that is integrated into the FTD product
Snort	Components of the Firepower product integrated into FTD
FMC	Firepower Management Center – Off-box GUI used to manage FTD devices (Configuration, reporting, monitoring, etc.). Formerly the Firesight Management Center or Defense Center.
FDM	Firepower Device Manager – Web-based, on-box management option for low to mid-range platforms
FXOS	Firepower Extensible Operating System – System that manages the hardware platforms for Firepower 9300, 4100, 3100, and 2100 series products
FCM	Firepower Chassis Manager – On-box GUI used to manage FXOS platforms (Logical device configuration, interface assignments, monitoring, etc.)



# Architecture Overview: Software Functions

# Introduction – What is Firepower Threat Defense?

- ASA and Firepower functionality wrapped into a single, unified image
- All processes run within single operating system
- Latest hardware platforms introduce Firepower Extensible Operating System (FXOS) as platform layer beneath the FTD application

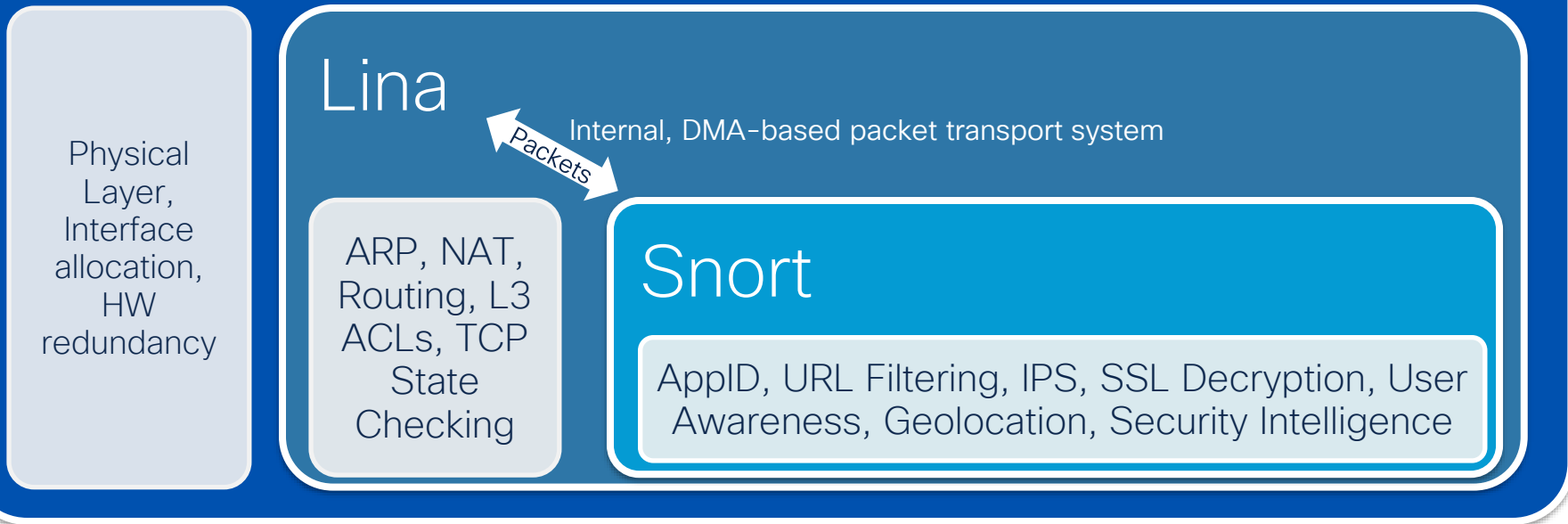


# Functional Overview – A Layered Approach

OSI Layer	Component	Examples
L1 - Physical	FXOS, 5500-X, Virtual platforms	Interface allocation, L1 configuration
L2 - Data Link	Lina (FXOS handles LACP on Firepower platforms – 1k, 2k ,3k, 4k, 9300)	Interface MAC Addressing, ARP
L3 - Network	Lina	IP Address assignment, Routing, NAT
L4 - Transport	Lina	TCP State checking, L4 ACLs
L5-7 - Session, Presentation, and Application Layers	Snort (Lina L7 inspection via MPF)	AppID, URL Filtering, IPS, SSL Decryption, User Awareness

# Firepower Threat Defense - Functional Diagram

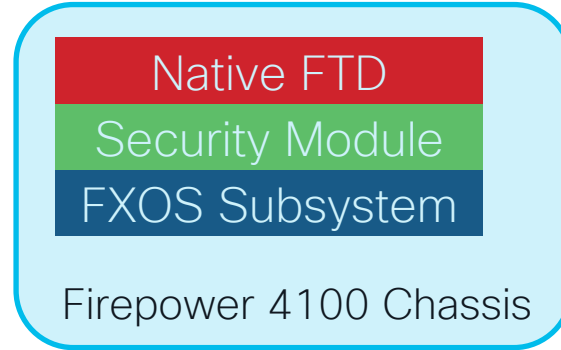
Platform (Virtual, 5500-X, FPR 1k, 2k\*, 3k, 4k, 9300)



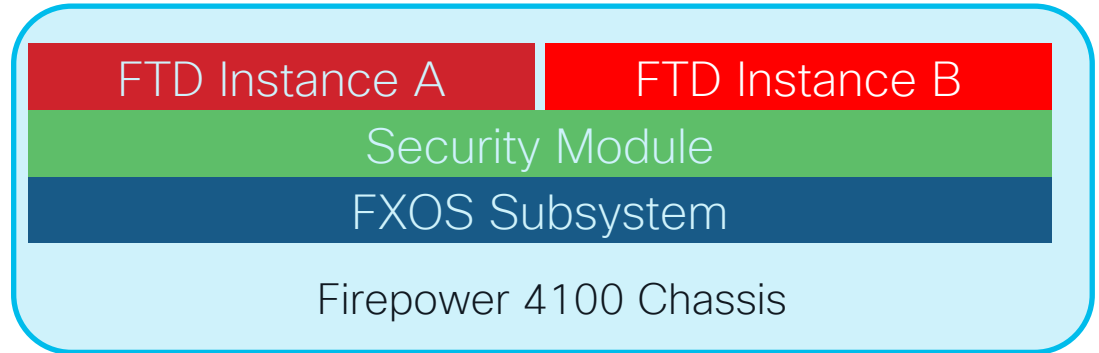


# FTD Deployment types in FXOS

## 1. Native Mode

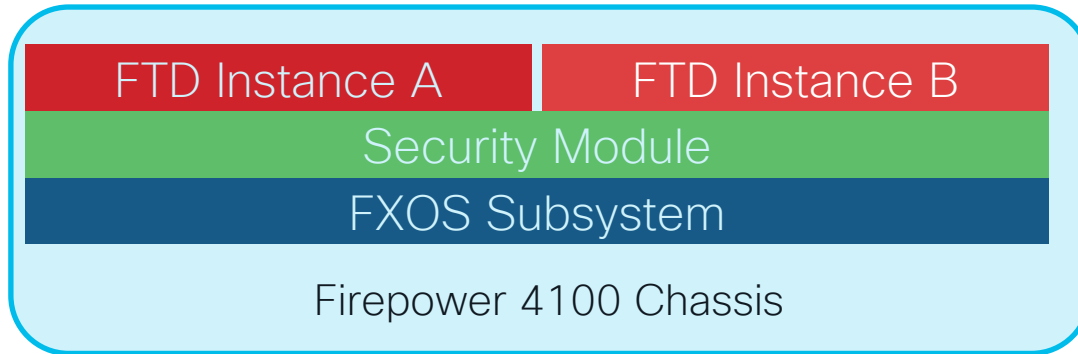


## 2. Multi-Instance Mode Supported only in FPR4100 and FPR9300



# Multi-Instance FTD on FXOS Platforms

- MI feature was released in FTD 6.3 (December 2018)
- Similar to ASA multi-context feature but implementation is different:
  - Leverage the container infrastructure within FXOS
  - Enables reboot/upgrade of individual instances without affecting other instances
  - Improved hardware resource separation since each instance has its own dedicated CPU cores, disk space, and memory





For your  
reference

# FTD – Navigating between the CLIs

FXOS (FPR 1k, 2k, 3k, 4k, 9300 platforms)

```
FPR9300#
```

Blade CLI (4100, 9300 platforms)



connect module 1 telnet



exit

```
Firepower-module1>
```

FTD Unified CLI (CLISH)



connect ftd



exit ('connect fxos' on FPR 1k, 2k, 3k)

```
>
```

expert

Expert shell (BASH)

```
admin@Firepower-module1:/opt/bootcli/cisco/cli/bin$
```

exit

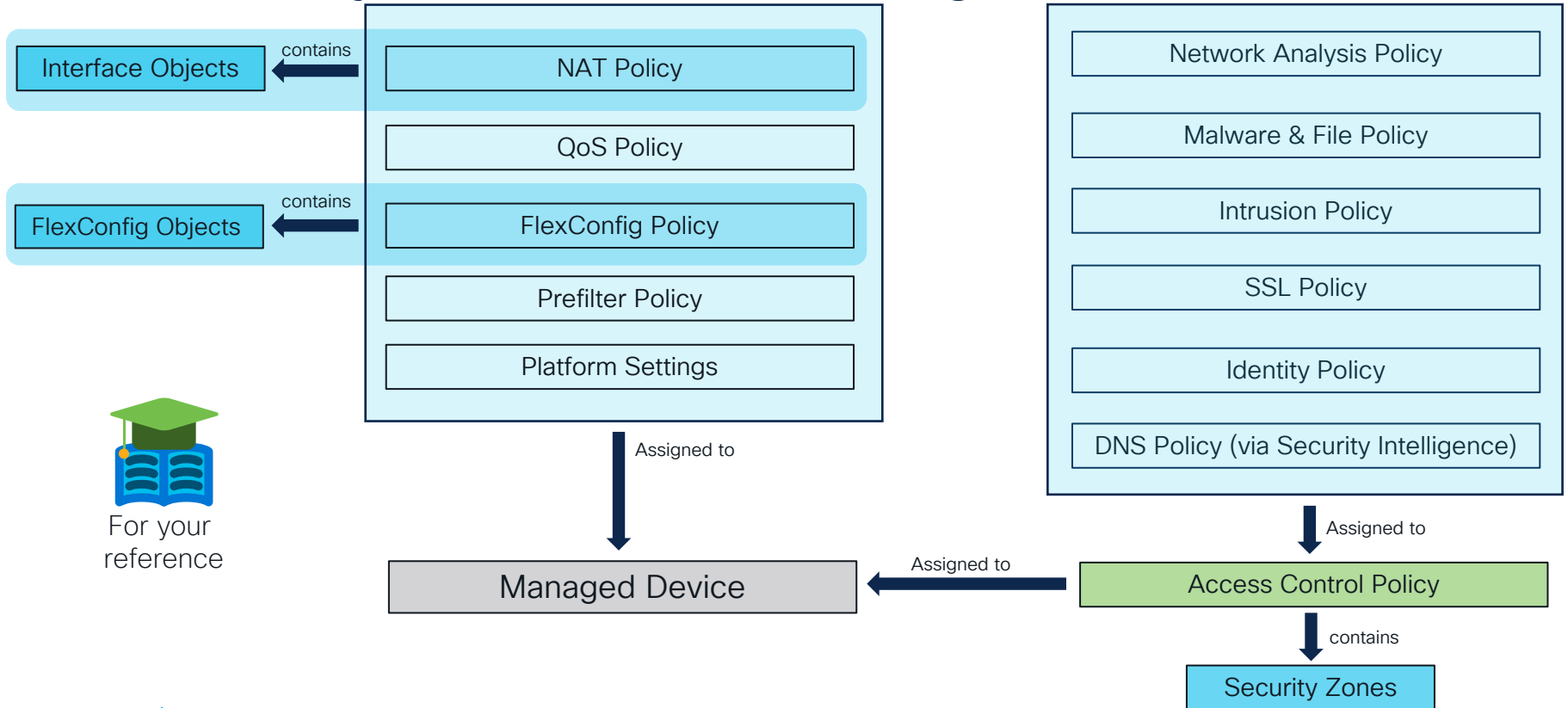
system support diagnostic-cli

Lina shell

```
firepower#
```

exit

# FMC – Object Relationship Diagram



For your reference

# Functional Overview – Physical Layer (L1)

- On FXOS platforms, interface allocation is handled via Firepower Chassis Manager (FCM) or the FXOS CLI.

The screenshot shows the Cisco Firepower Chassis Manager (FCM) web interface. The top navigation bar includes Overview, Interfaces (selected), Logical Devices, Security Modules, and Platform Settings. The main area displays three network modules: Network Module 1, Network Module 2, and Network Module 3. Network Module 1 has 8 ports, with ports 1, 2, 5, and 6 highlighted in red. Network Module 2 has 8 ports, with port 2 highlighted in red. Network Module 3 has 8 ports, with port 3 highlighted in red. Below the network module diagrams is a table titled 'All Interfaces' with columns for Interface, Type, Admin Speed, Operational Speed, Application, Operation State, and Admin State. The table lists several interfaces, including MGMT, Port-channel1, Ethernet1/1, Ethernet1/2, Port-channel3, Ethernet1/5, Ethernet1/6, and Port-channel48.

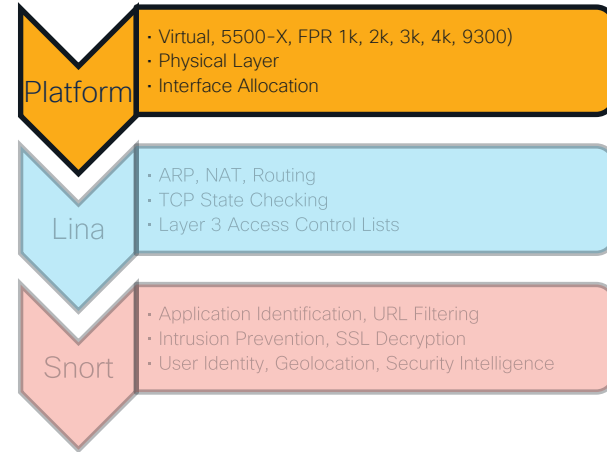
Interface	Type	Admin Speed	Operational Speed	Application	Operation State	Admin State
MGMT	Management					Enabled
Port-channel1	data	1gbps	1gbps	FTD	failed	Enabled
Ethernet1/1					individual	
Ethernet1/2					down	
Port-channel3	data	1gbps	indeterminate	FTD	failed	Enabled
Ethernet1/5					down	
Ethernet1/6					down	
Port-channel48	cluster	10gbps	indeterminate		failed	Enabled

# Functional Overview – Physical Layer (L1)

Viewing interface statistics in FXOS CLI:

```
FPR9300-A# scope eth-uplink
FPR9300-A /eth-uplink # scope fabric a
FPR9300-A /eth-uplink/fabric # show interface detail

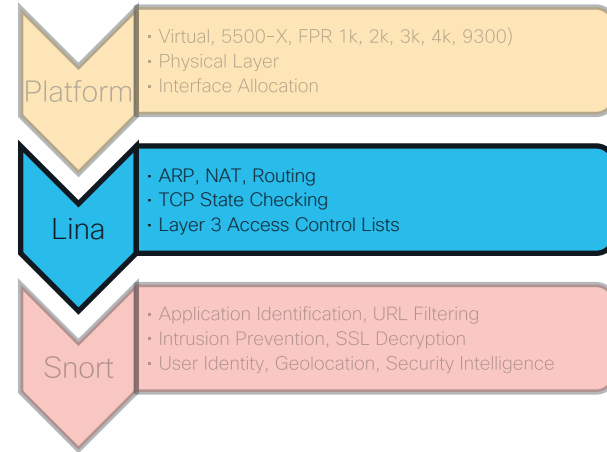
Interface:
  Port Name: Ethernet1/3
  ...
FPR9300-A /eth-uplink/fabric # scope interface 1 3
FPR9300-A /eth-uplink/fabric/interface # show stats
...
Ether Rx Stats:
  Time Collected: 2017-04-17T23:45:33.906
  Monitored Object: sys/switch-A/slot-1/switch-ether/port-3/rx-
stats
  Suspect: No
  Total Packets (packets): 8968254
  Total Bytes (bytes): 1798297716
  Unicast Packets (packets): 1098012
  Multicast Packets (packets): 2480578
  Broadcast Packets (packets): 5389664
```



# Functional Overview – Data/Network Layer (L2/3)

You can see L2 and L3-related interface information in the Unified CLI:

```
> show interface Ethernet1/3
Interface Ethernet1/3 "diagnostic", is up, line protocol is up
Hardware is EtherSVI, BW 1000 Mbps, DLY 1000 usec
  MAC address b0aa.772f.849c, MTU 1500
  IP address 10.10.1.1, subnet mask 255.255.255.0
Traffic Statistics for "diagnostic":
  4380985 packets input, 201525318 bytes
  0 packets output, 0 bytes
  162 packets dropped
  1 minute input rate 9 pkts/sec, 437 bytes/sec
  1 minute output rate 0 pkts/sec, 0 bytes/sec
  1 minute drop rate, 0 pkts/sec
  5 minute input rate 9 pkts/sec, 446 bytes/sec
  5 minute output rate 0 pkts/sec, 0 bytes/sec
  5 minute drop rate, 0 pkts/sec
Management-only interface. Blocked 0 through-the-device packets
```

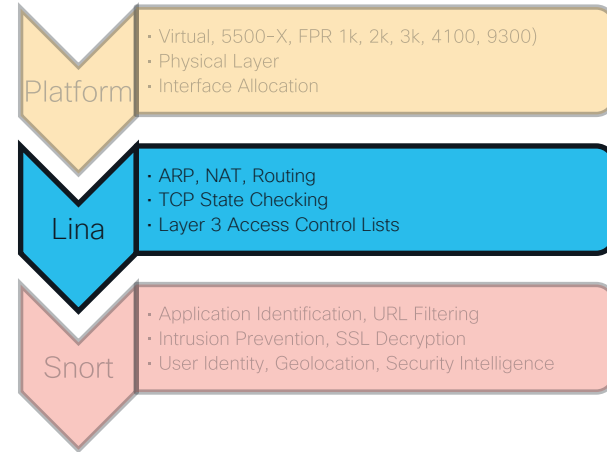


\*Note that the above interface is a management-only interface

# Functional Overview – Network Layer (L3)

You can also view NAT configuration and active routes in the Unified CLI:

```
> show running-config nat
nat (inside,outside) source dynamic INSIDE_NETS interface
!
object network SRV-10.10.1.100-REAL
  nat (inside,outside) static SRV-10.10.1.100-GLOBAL
!
> show route
...
S*      0.0.0.0 0.0.0.0 [1/0] via 172.18.249.1, outside
C      169.254.1.0 255.255.255.252 is directly connected, nlp_int_tap
L      169.254.1.1 255.255.255.255 is directly connected, nlp_int_tap
>
```



All legacy ASA show and debug commands are still available in FTD via the `> system support diagnostic-cli` command



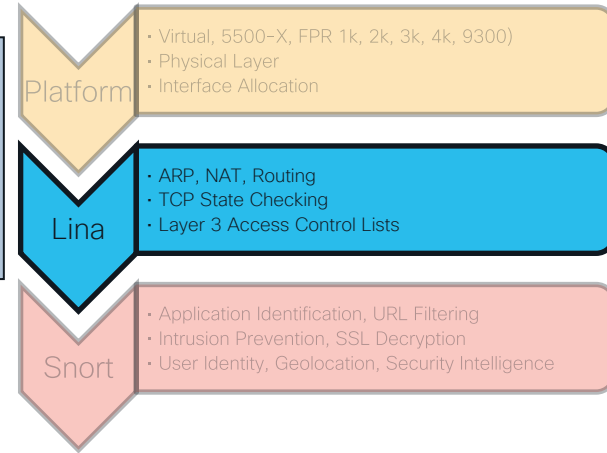
# Functional Overview – Network/Transport (L4)

TCP state and L3/L4 ACL checking are performed by the Lina process

```
> show conn protocol tcp
165 in use, 54084 most used

TCP outside 10.106.45.60:443 inside38 14.38.104.110:56946, idle 0:00:18..
TCP outside 108.171.133.146:8080 inside38 14.38.104.1:25148, idle 0:00:03..
TCP outside 108.171.133.146:8080 inside38 14.38.104.1:13080, idle 0:00:21..
>
```

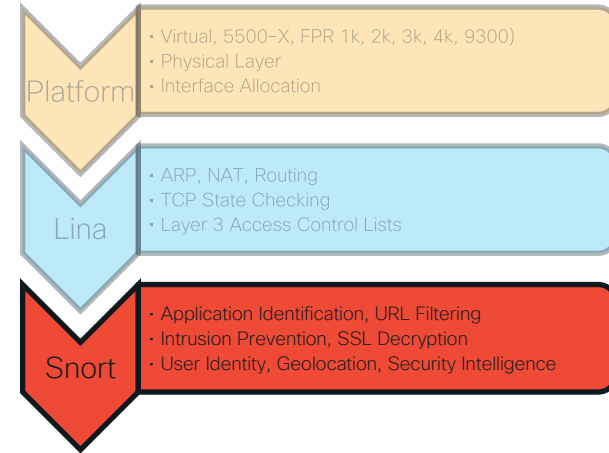
```
> show running-config access-list
access-list CSM_FW_ACL_ remark rule-id 2684445405: PREFILTER POLICY: Default Prefilter Policy_1
access-list CSM_FW_ACL_ remark rule-id 268444672: ACCESS POLICY: FTD-ACPolicy-201703230950 - Default/1
access-list CSM_FW_ACL_ remark rule-id 268444672: L7 RULE: from_outside_#1
access-list CSM_FW_ACL_ advanced permit udp ifc outside 10.2.2.0 255.255.255.0 host 10.1.1.100 eq syslog
rule-id 268444672
...
```



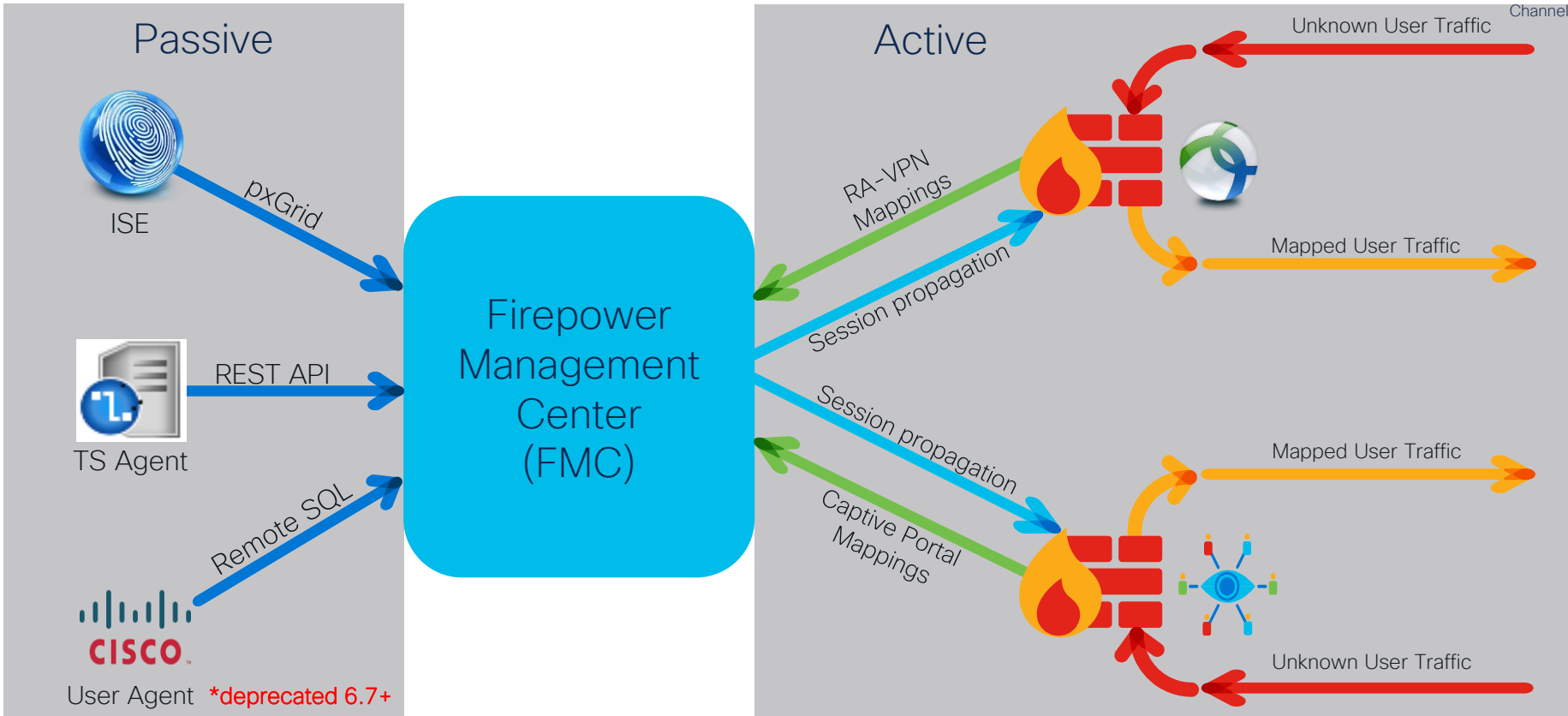
# Functional Overview – Upper Layers (5-7)

Short-handled functions that occur at upper OSI layers:

- Intrusion Prevention System (IPS)
- App Detection and OpenAppID
- URL Filtering
- SSL/TLS Decryption
- User Identity Awareness
- File and malware inspection



# Identity Architecture & Overview



# Architecture Overview: CPU and Memory Allocation

# FTD CPU and Memory Allocation

- CPU and memory are allocated to Lina and Snort via the use of Linux cgroups
- This resource pool (cgroup) separation limits scope of problem impact
- Troubleshooting approach depends on where issue resides

Example of Lina and Snort CPU allocations on a Firepower 9300

```
firepower# show kernel cgroup-controller cpuset | begin lina
group "restricted/lina" ← Lina
cpuset.cpus: 1-8,18-25,37-44,54-61
cpuset.mems: 0-1
tasks:
    12507  12794  12803  12804
    12805  15917  15918  15943

group "restricted/qemu" ← Snort
cpuset.cpus: 9-17,26-35,45-53,62-71
cpuset.mems: 0-1
```

# Lina Memory – Overview

- Lina memory is broken into two categories: Shared memory and DMA memory

```
firepower# 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
```

- Use CISCO-ENHANCED-MEMPOOL-MIB.my for accurate SNMP counters
- Free memory may not recover immediately after conn spike due to caching
- Connections, Xlates and ACL configuration are top users of shared memory

# Lina Memory Blocks (Direct Memory Access)

- DMA memory involves fixed-size blocks allocated at startup
- Used for packet processing, VPN, etc.

firepower# 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
9344	2000	2000	2000
16384	102	102	102
65536	16	16	16

firepower#

Current number of free blocks available

1550, 2048, and 9344 byte blocks are used for processing Ethernet frames

When DMA memory for a specific block size runs low, the following syslog will be generated for the specific block size:

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

# Lina CPU Utilization 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

```
> show process cpu-usage sorted non-zero
```

PC	Thread	5Sec	1Min	5Min	Process
0x08dc4f6c	0xc81abd38	14.4%	8.2%	8.0%	<b>SNMP Notify Thread</b>
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.

If you have high CPU utilization for a generic process such as DATAPATH, contact the TAC as there are more granular CPU profiling tools available for deeper investigation



# Snort, Lina, and the Firepower ecosystem

- Many processes run on Linux to support event collection and other management, including:

Process	Primary Purpose
Lina	ASA-like functions: L4 ACLs, ALG, Routing, Failover, Clustering, etc
Snort	Inspects traffic and writes events to unified log files
SFDataCorrelator	Read unified logs written by snort, and send events to FMC
sftunnel	Manage an encrypted connection back to the FMC over TCP/8305
ids_event_alerter	Sends syslogs and SNMP traps from sensor for intrusion events

- Process status can be verified with: > **pmtool status**
- Standard Linux troubleshooting tools, such as “**top**,” can be used to verify CPU and memory

# Expert Mode - CPU Utilization by Processes

Open “top” program from BASH (Sorting by CPU is the default)

```
> expert
admin@firepower:~$ top
```

Cpu(s): 15.3%us, 5.8%sy, 0.0%ni, 78.4%id, 0.0%wa, 0.0%hi, 0.5%si, 0.0%st  
Mem: 12321960k total, 5605756k used, 6716204k free, 148992k buffers  
Swap: 3998716k total, 780k used, 3997936k free, 1222064k cached

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
12221	root	0	-20	1896m	299m	75m	S	100	2.5	2733:37	lina
22420	root	20	0	618m	8048	2980	S	42	0.1	1539:57	sftunnel
14777	root	20	0	2185m	60m	12m	S	0	0.5	8:11.23	SFDataCorrelator
25979	root	20	0	1893m	347m	12m	S	0	2.9	2:15.42	snort

Processes sorted by CPU

- Lina handles its own resources. Disregard high CPU and memory readings for Lina in “top”
- Occasional high CPU for Snort is determined by current flow

# Expert Mode - Memory Utilization by Processes

```
> expert
admin@firepower:~$ top
```

1. Open “top” program
2. Type “shift + f” to choose sorting field
3. Type “n” to select resident memory

```
Current Sort Field: N for window 1:Def
Select sort field via field letter
k: %CPU           = CPU usage
l: TIME           = CPU Time
m: TIME+         = CPU Time, hundredths
* N: %MEM         = Memory usage (RES)
o: VIRT          = Virtual Image (kb)
```

Tasks: 465 total, 1 running, 464 sleeping, 0 stopped, 0 zombie  
Cpu(s): 41.6%us, 0.3%sy, 0.0%ni, 58.1%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st  
Mem: 132166192k total, 43796884k used, 86636864k free, 252k buffers  
Swap: 7810780k total, 0k used, 7810780k free, 1732192k cached

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
12506	root	0	-20	26.1g	1.1g	643m	S	1993	0.8	97328:59	lina
11949	root	1	-19	7813m	671m	37m	S	2	0.5	6:15.66	snort
12902	root	20	0	4129m	68m	16m	S	2	0.1	41:54.55	SFDataCorrelator

Processes sorted by resident memory

# Expert Mode - Memory Management Example

- Snort is the primary memory consumer, and will use more memory over time
- Low system memory is not necessarily a sign of a problem

Round numbers used to  
simplify example

```
"System" cgroup
Limit: 5 GB
```

Memory	Process
1 GB	lina
1 GB	SFDataCorrelator
1 GB	Database
1 GB	DiskManager
1 GB	ids_event_alerter

```
"Detection" cgroup
Limit: 10 GB
```

Memory	Process
2 GB	snort
2 GB	snort
2 GB	snort
2 GB	snort
2 GB	snort
2 GB	snort

Errors in /var/log/messages

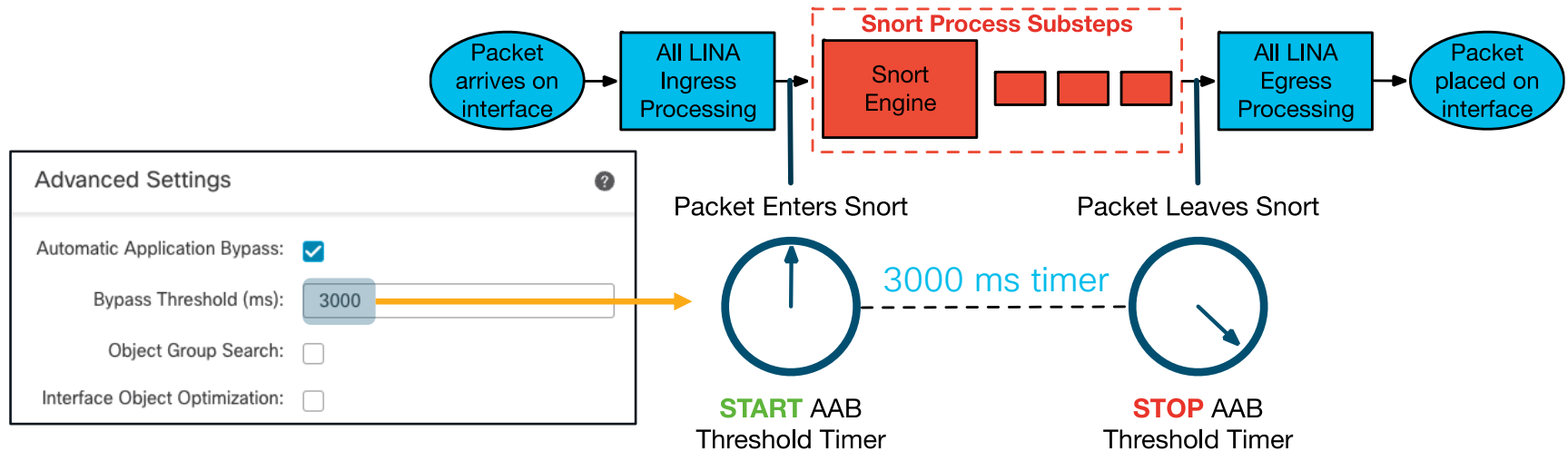
```
kernel: SFDataCorrelator invoked oom-killer: gfp_mask=0xd0, order=0, oom_adj=0
```

```
kernel: Task in /System killed as a result of limit of /System
```

```
Out of memory: Kill process 4715 (snort) score 258 or sacrifice child
```

- Snort is protected from low-memory issues caused by processes in other cgroups

# Snort - Automatic Application Bypass (AAB)



- AAB is a per packet timer for snort
- A snort instance is killed if a packet fails to egress before the threshold
- A snort core file is collected for root cause analysis
- The process manager will respawn snort
- Configured under **Devices > Device Management > Advanced Settings**
- Do **not** go below 3000 milliseconds threshold unless recommended by TAC

# Snort - Intelligent Application Bypass (IAB)

- IAB is a performance optimization tool for elephant flows
- Invoked in a simple 2-step process:
  1. Does snort exceed the "Inspection Performance Thresholds" (high CPU, % dropped traffic, etc)?
  2. If yes, then dynamically Trust flows which match "Flow Thresholds" (bytes/sec, packets/flow, etc)
- Configured under Access Control Policy > Advanced tab

The screenshot shows the 'Intelligent Application Bypass Settings' dialog box. At the top right is a help icon (?). The 'State' is set to 'Off' in a dropdown menu. Below it, the 'Performance Sample Interval (seconds)' is set to '5' in a text input field. The 'Bypassable Applications and Filters' section has two radio buttons: '0 Applications/Filters' (unselected) and 'All applications including unidentified applications' (selected). The 'Inspection Performance Thresholds' section has a 'Configure' link. The 'Flow Bypass Thresholds' section also has a 'Configure' link. At the bottom, there is an information icon (i) with the text: 'Only 'Drop Percentage' and 'Flow Velocity' settings are applicable for Snort 3 devices.' At the very bottom of the dialog are three buttons: 'Revert to Defaults', 'Cancel', and 'OK'.

# Expert Mode - Core Files

- If a process on Linux exits unexpectedly, a core file may be written to the file system

FTD on FPR 1k, 2k, 3k, 4k, FP9300	FTD on ASA and Virtual Platforms (VMware, KVM, AWS, Azure)
<ul style="list-style-type: none"><li>Cores written to <code>/opt/cisco/csp/cores/</code></li><li>Core automatically compressed and moved to <code>/ngfw/var/data/cores/</code></li></ul>	<ul style="list-style-type: none"><li>Cores written uncompressed to <code>/ngfw/var/common/</code></li></ul>
<code>core.snort.6.5373.1496879772.gz</code>	<code>core_1496879772_sensor_snort_6.5373</code>

Process name

POSIX  
kill signal

Process  
ID

Unix Epoch Timestamp  
(Secs since 1970-Jan-01)

Hostname

# Expert Mode - Disk Management

- The DiskManager process manages collections of files called “silos”
- If space is low, DiskManager will prune each silo based on a preconfigured threshold

```
> show disk-manager
```

<b>Silo</b>	<b>Used</b>	<b>Minimum</b>	<b>Maximum</b>
Temporary Files	0 KB	584.291 MB	2.282 GB
Backups	0 KB	4.565 GB	11.412 GB
Updates	0 KB	6.847 GB	17.118 GB
Archives & Cores & File Logs	0 KB	4.565 GB	22.824 GB
RNA Events	0 KB	4.565 GB	18.259 GB
File Capture	0 KB	11.412 GB	22.824 GB
Connection Events	0 KB	413.320 MB	826.642 MB
IPS Events	0 KB	13.694 GB	34.236 GB
[lines_removed]			



# Expert Mode - Disk Management

- The Lina file system is accessible from expert mode via /mnt/disk0

```
# Create a capture from the unified CLI
> capture CAPTURE match ip any host 8.8.8.8

# Enter the diagnostic (lina-only) CLI
> system support diagnostic-cli
firepower# copy /pcap capture:CAPTURE disk0:CAPTURE.pcap

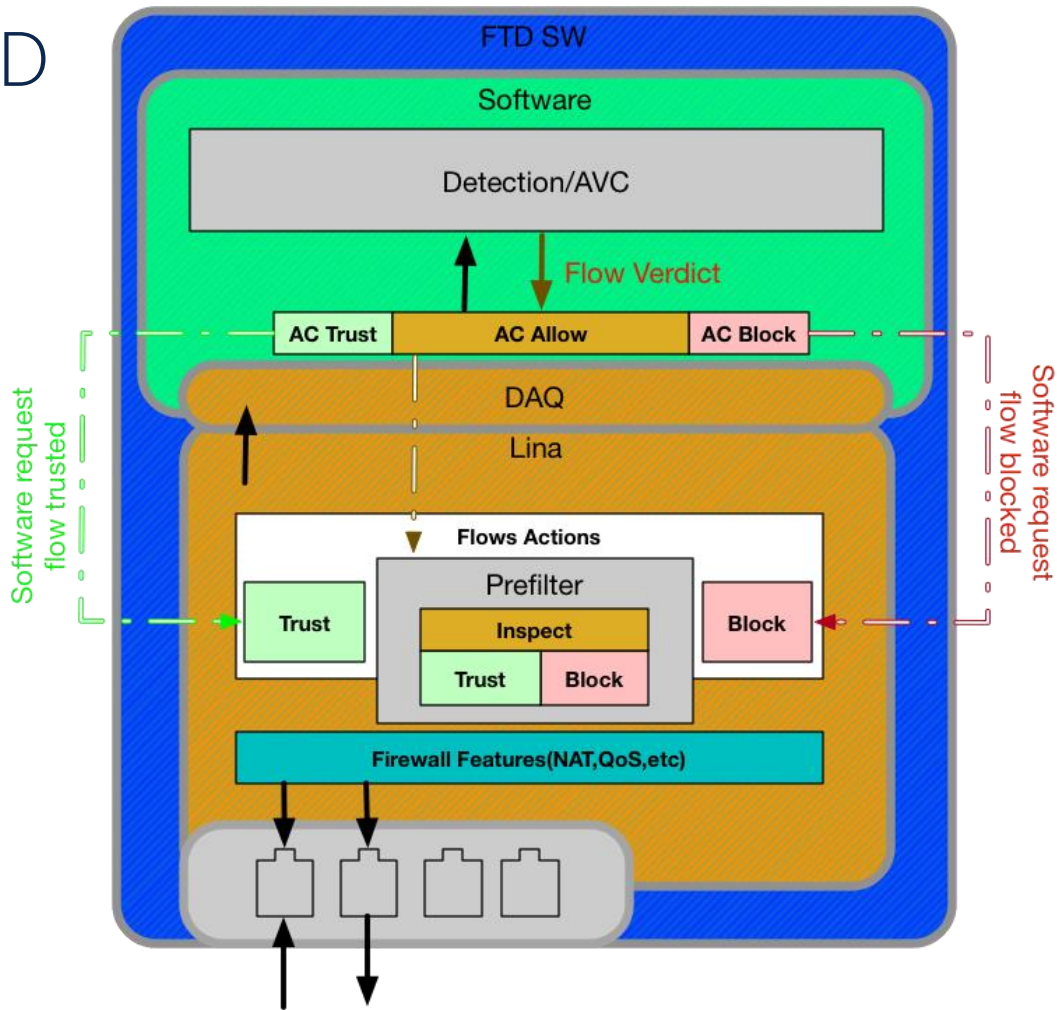
# Enter expert mode and browse to /mnt/disk0
> expert
admin@FPR4100:/mnt/disk0 $ ls
CAPTURE.pcap
[lines_removed]
```

# The Path of the Packet (Platform Architecture)



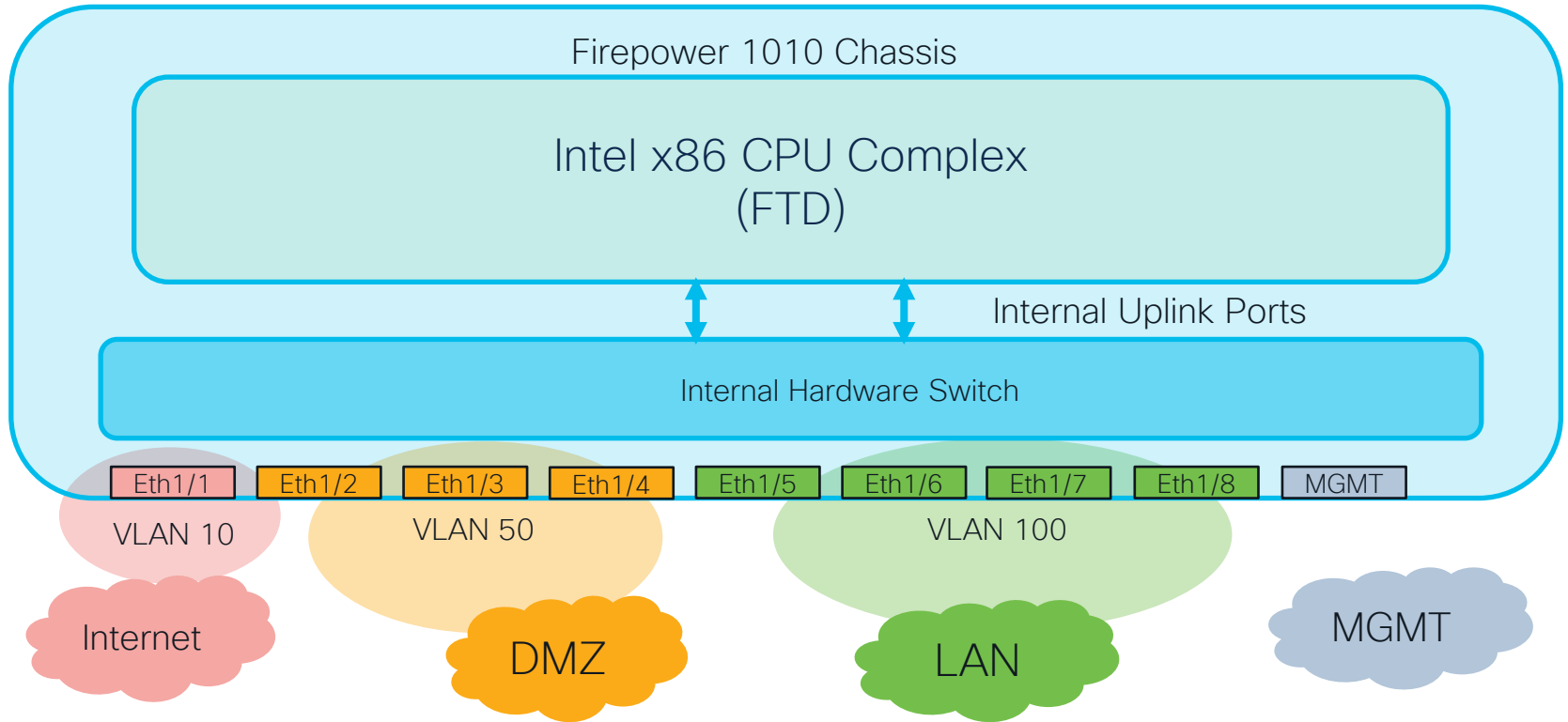
# Virtual FTD

VMWare  
AWS  
Azure  
KVM  
Hyper-V  
FPR 1k  
FPR 3k

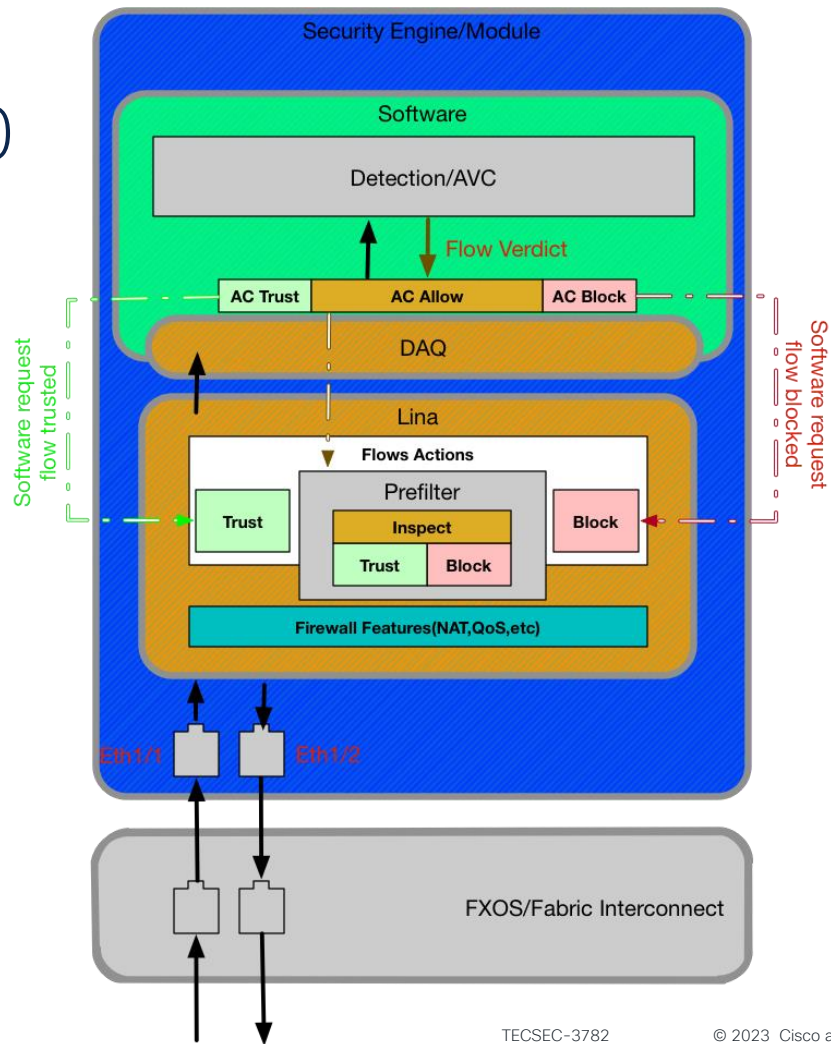


# Firepower 1010 - L2 Switching Overview

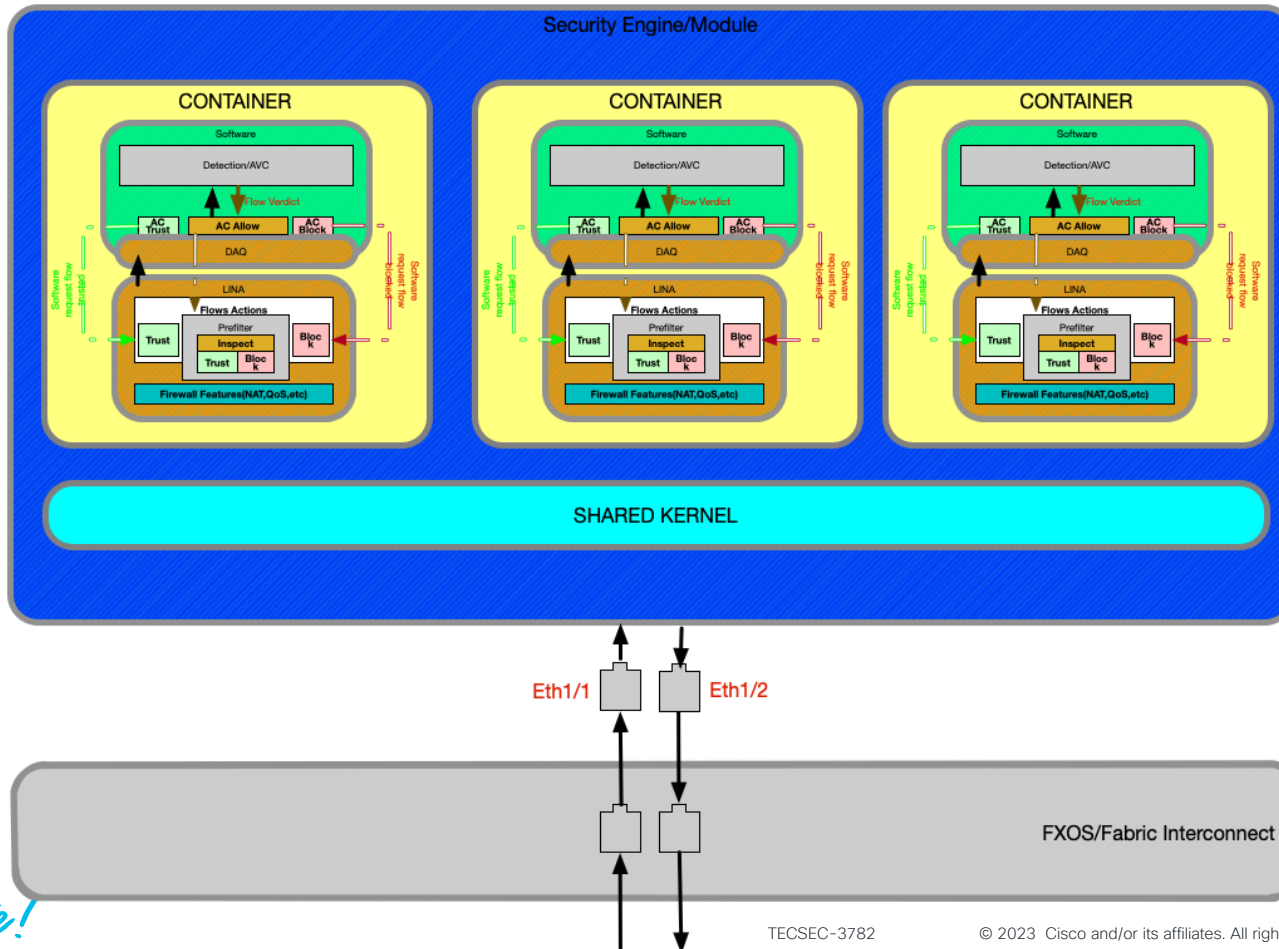
- New in 6.5 - Eliminates the need for an external switch in SOHO environments



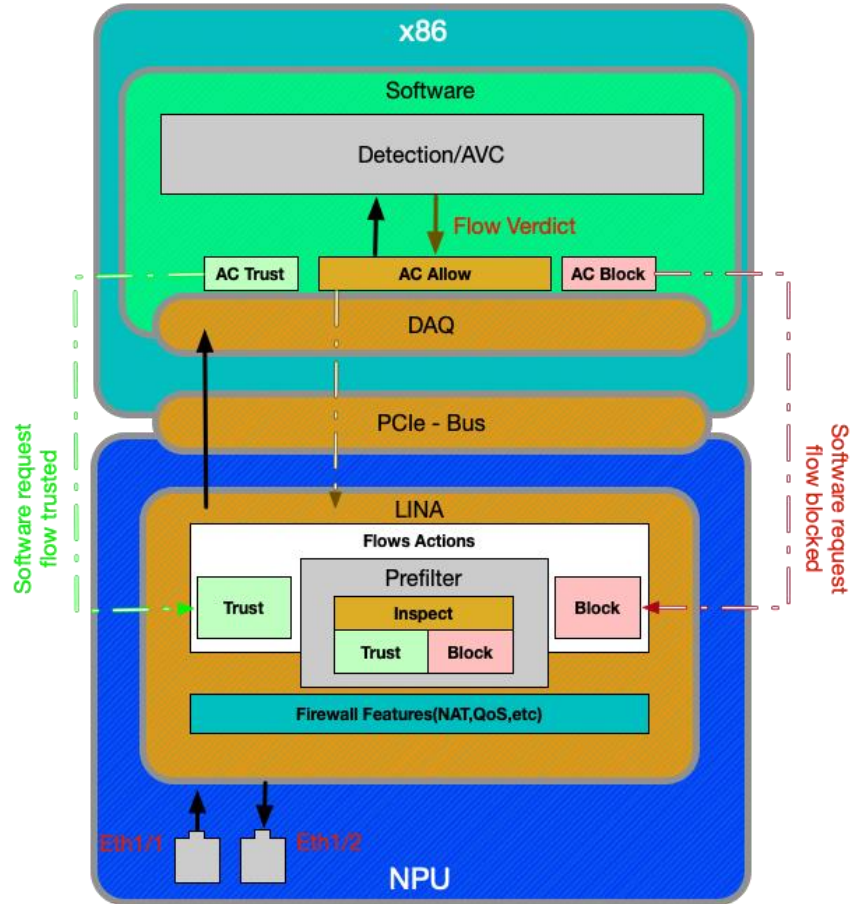
# Firepower 4100 & 9300



# Multi-Instance architecture overview(9300/4100)



# Firepower 2100



# The Path of the Packet (Software / Logical Flow)





# Understanding Packet Flow

Effective troubleshooting requires an understanding of the packet path in network

1. Attempt to isolate the problem down to a single device
2. Perform a systematic walk of the packet through device to identify problem

For problems relating to FTD, always

- Determine the flow: Protocol, Source IP, Destination IP, Source Port, Destination Port
- Determine the logical (named) interfaces through which the flow passes

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

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

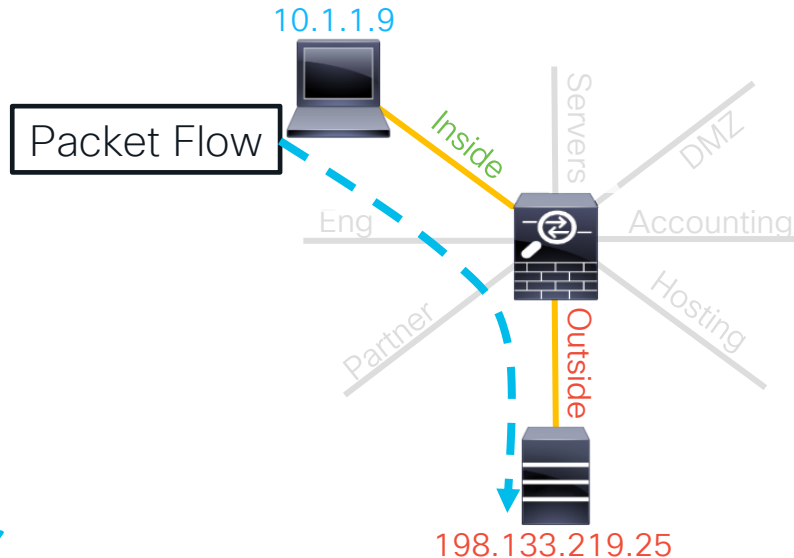
# Example Flow

- TCP Flow

- Source IP: 10.1.1.9                      Source Port : 11030
- Destination IP: 198.133.219.25      Destination Port: 80

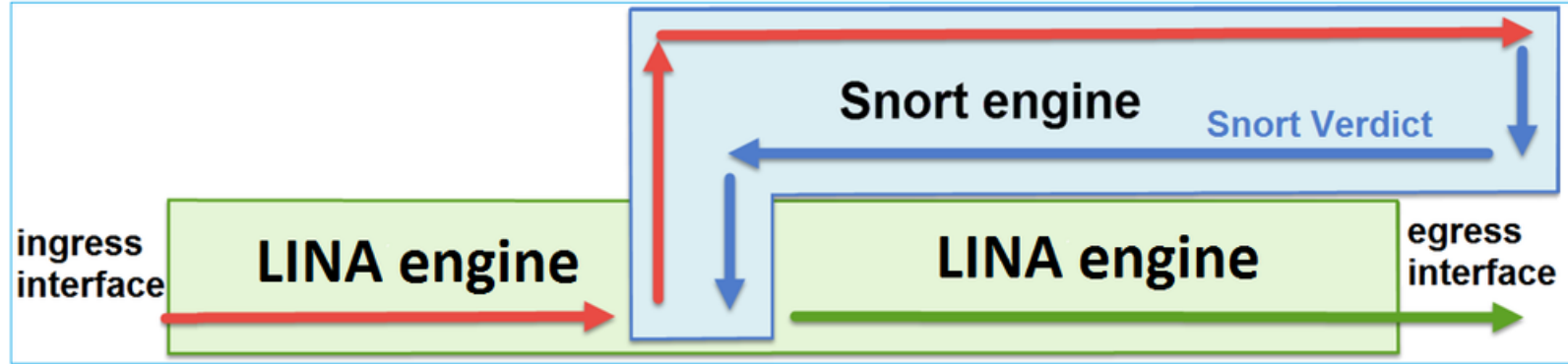
- Interfaces

- Source: **Inside**                                      Destination: **Outside**



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

# FTD Packet Processing – The Big Picture

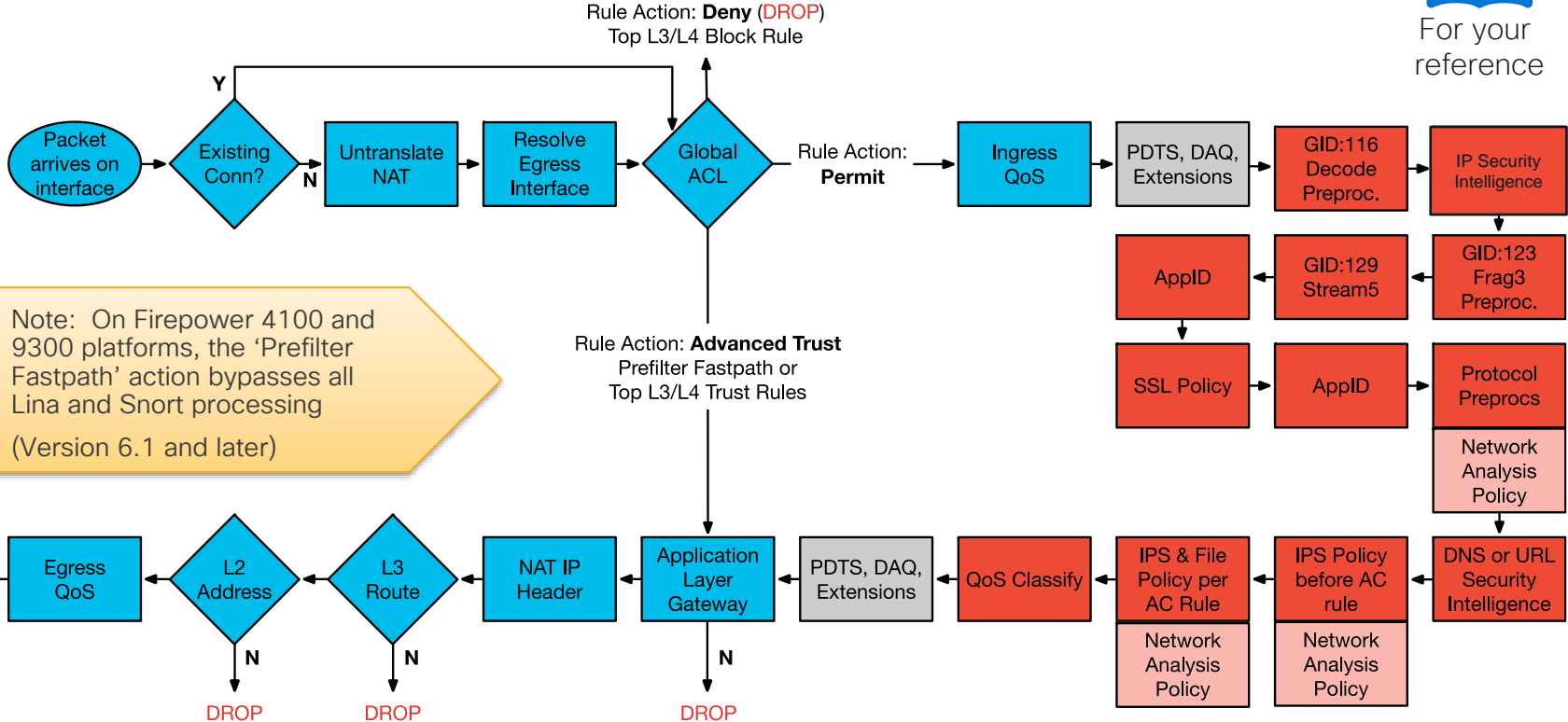


1. Packet enters the ingress interface, and it is handled by the LINA engine
2. If the policy dictates so the packet is inspected by the Snort Engine.
3. Snort Engine returns a verdict for the packet
4. Lina Engine drops or forwards the packets based on Snort's verdict

# Reference Slide: Routed FTD Path of Packet



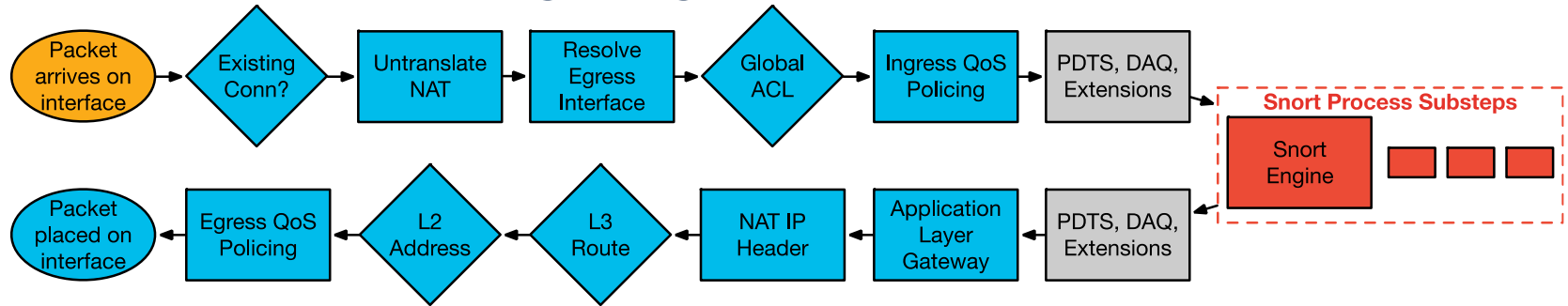
For your reference



LINA ASA Engine = BLUE

Snort Engine = RED

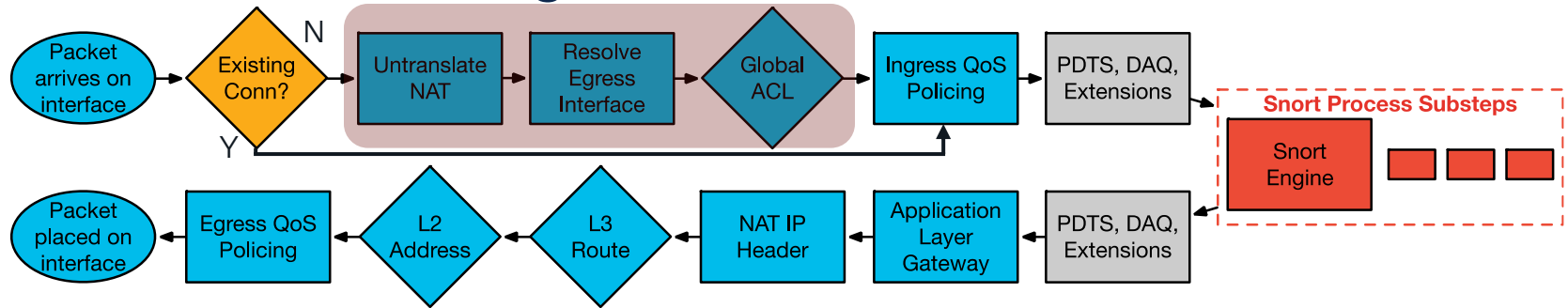
# Packet Processing: Ingress interface



- 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
- **Overflow** counter indicates packet drops (usually packet bursts)

```
> show interface outside
Interface GigabitEthernet0/3 "outside", is up, line protocol is up
  Hardware is i82546GB rev03, BW 1000 Mbps, DLY 10 usec
  [...]
    IP address 148.167.254.24, subnet mask 255.255.255.128
    54365986 packets input, 19026041545 bytes, 0 no buffer
    Received 158602 broadcasts, 0 runts, 0 giants
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  [...]
    input queue (blocks free curr/low): hardware (255/230)
    output queue (blocks free curr/low): hardware (254/65)
```

# Packet Processing: Locate Connection



- Check for existing connection in conn table

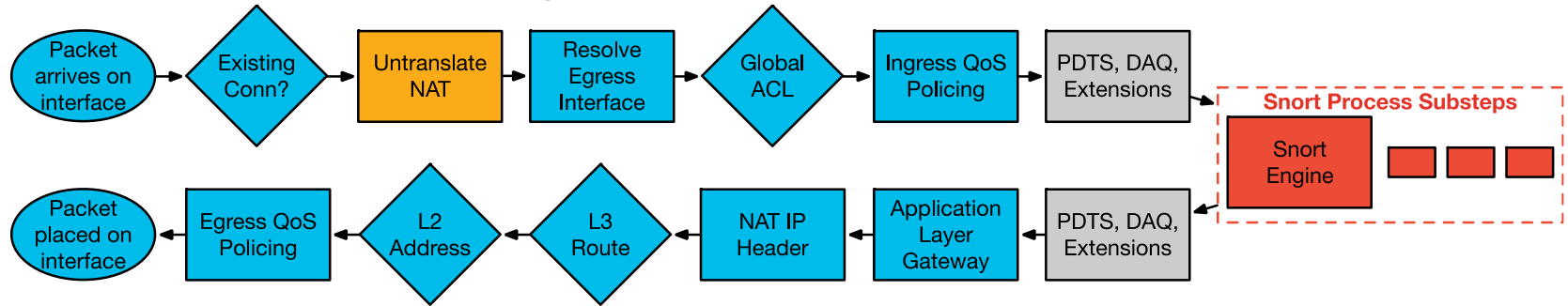
```
> show conn
TCP out 198.133.219.25:80 in 10.1.1.9:11030 idle 0:00:04 Bytes 1293 flags UIO
```

- 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
```

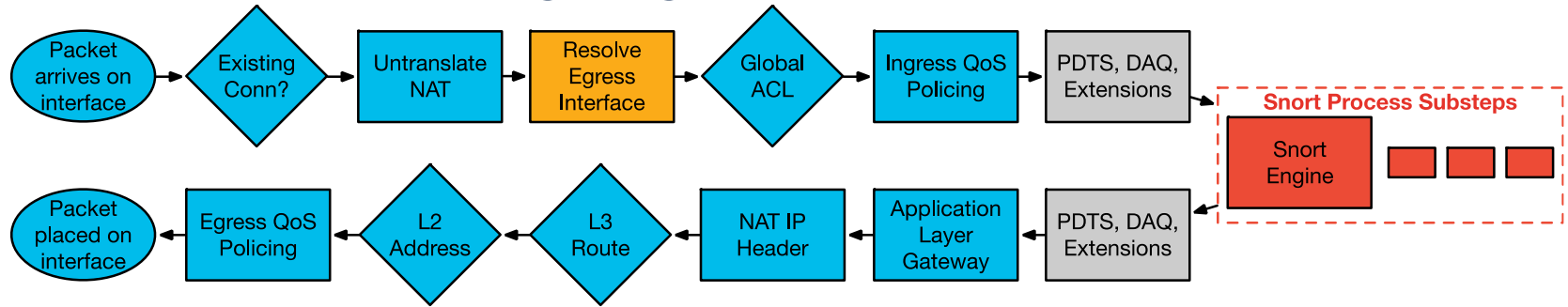
If connection entry exists, bypass ACL check and process in Lina fastpath

# Packet Processing: NAT Un-Translate

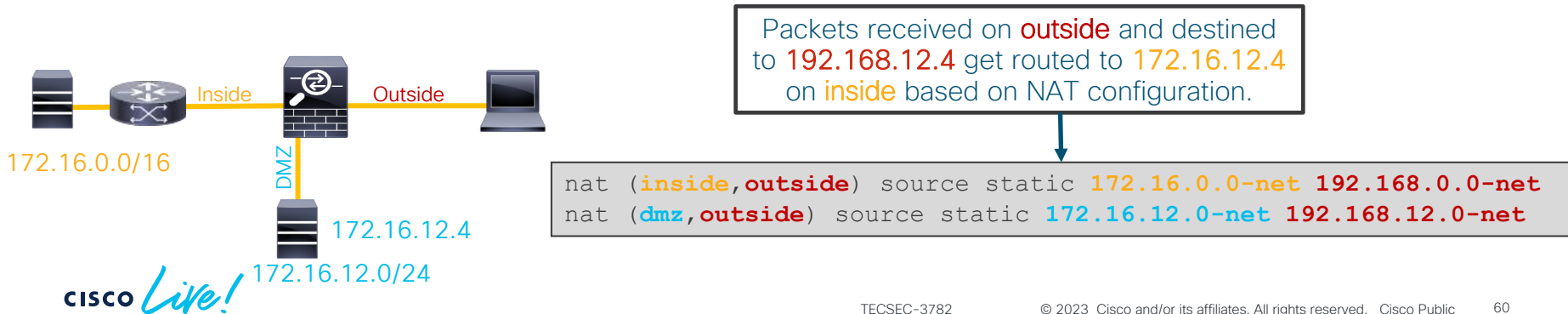


- Incoming packet is checked against NAT rules
- Packet is un-translated first, before ACL check
- NAT rules that translate the destination of the packet can override the routing table to determine egress interface (NAT divert)
  - Could also override policy-based routing (PBR)

# Packet Processing: Egress Interface

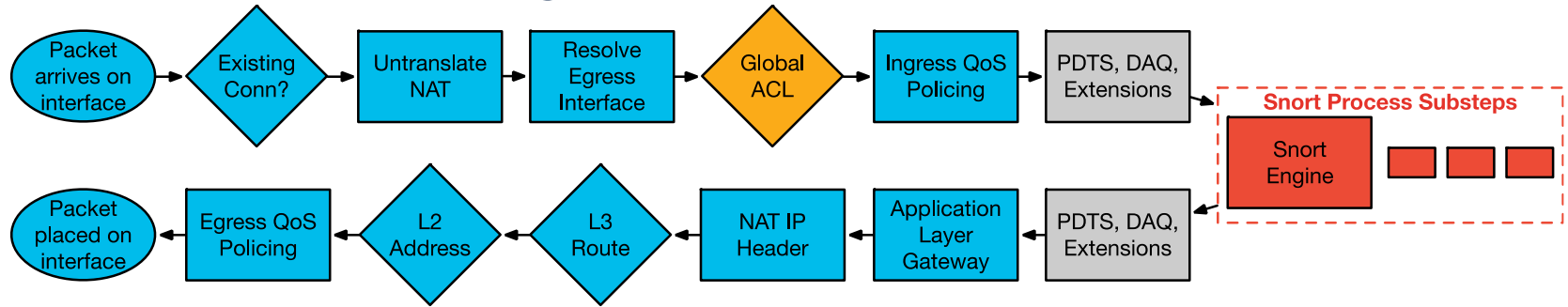


- Egress interface is determined **first** by translation rules or existing conn entry
- If NAT does not divert to the egress interface, the global routing table is consulted to determine egress interface





# Packet Processing: Global 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

```
> show access-list
...
CSM_FW_ACL_line 5 advanced permit tcp any any rule-id 9998 (hitcnt=5) 0x52c7a066

> show running-config access-group
access-group CSM_FW_ACL_global
```

# Packet Processing: Global ACL Check

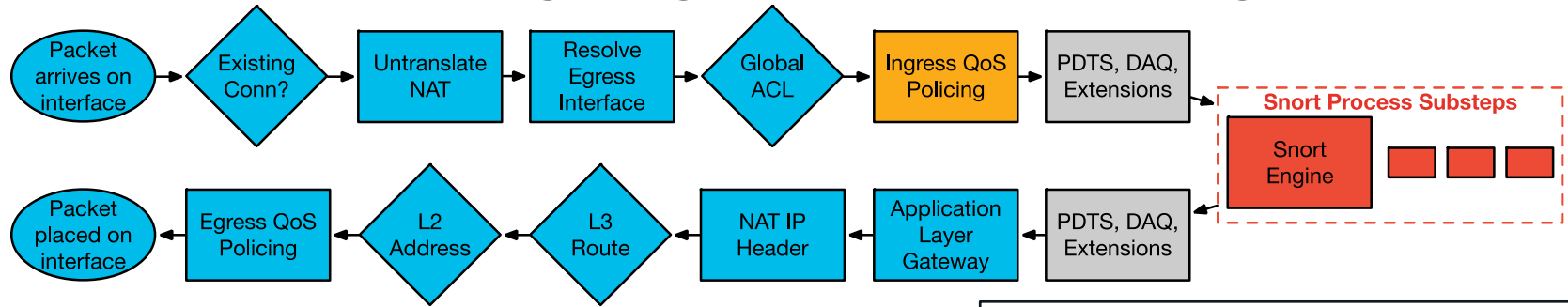
- All L4 access control entries are in one global ACL
- Prefilter Fastpath rules skip snort and show up as “**Advanced Trust**” in Lina global ACL

Rules												
+ Add Tunnel Rule + Add Prefilter Rule <input type="text" value="Search Rules"/>												
#	Name	Rule Type	Source Interface O...	Destination Interface O...	Source Networks	Destination Networks	Source Port	Destination Port	VLAN Tag	Action	Tunnel Zone	
1	Fastpath-rule	Prefilter	any	any	10.1.2.3	any	any	any	any	Fastpath	na	0

```
> show running-config access-group
access-group CSM_FW_ACL_ global

> show access-list
[lines_removed]
access-list CSM_FW_ACL_ line 1 remark rule-id 268435484: PREFILTER POLICY: FPR4100_Prefilter
access-list CSM_FW_ACL_ line 2 remark rule-id 268435484: RULE: Fastpath-rule
access-list CSM_FW_ACL_ line 3 advanced trust ip host 10.1.2.3 any rule-id 268435484 event-
log flow-end (hitcnt=0) 0x98824a05
```

# Packet Processing: Ingress QoS Policing

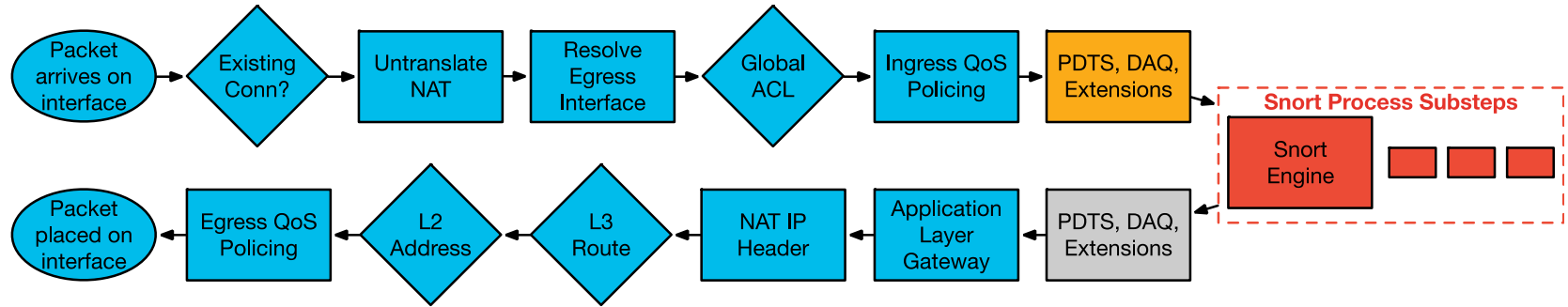


- QoS policing is enforced within the Lina process

```
> system support diagnostic-cli  
  
firepower# show service-policy interface inside  
Interface inside:  
  Service-policy: policy_map_inside  
  Flow-rule QoS id: 268435467  
  Input police Interface inside:  
    cir 1000000 bps, bc 31250 bytes
```

The screenshot shows the 'Add Rule' configuration page. The rule name is 'QoSRule1' and it is enabled. The 'Apply QoS On' dropdown is set to 'Interfaces in Source Interface Ot'. Under 'Traffic Limit Per Interface', both 'Download Limit' and 'Upload Limit' are set to 'Unlimited'. The page includes tabs for 'Interface Objects', 'Networks', 'Users', 'Applications', 'Ports', 'URLs', and 'SGT/ISE'. Below these are search fields for 'Available Realms' and 'Available Users', and a list of 'Special Identities'.

# Packet Processing: Packet Data Transport System



The Packet Data Transport System (PDTs) sends packets to Snort after initial Lina inspections

```
show asp inspect-dp snort
```

Displays conns and packets sent to each snort instance and process ID, as well as snort status

```
show asp inspect-dp snort counters summary
```

Display frames, bytes, and conns for snort instances

```
show asp inspect-dp snort queues
```

Display rx and tx queue utilization for snort instances

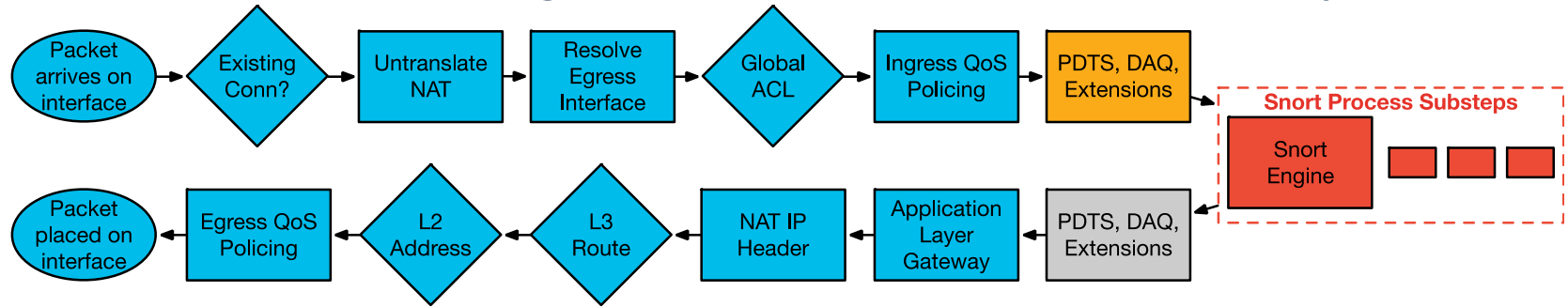
```
clear asp inspect-dp snort
```

Clear all of the above PDTs counters

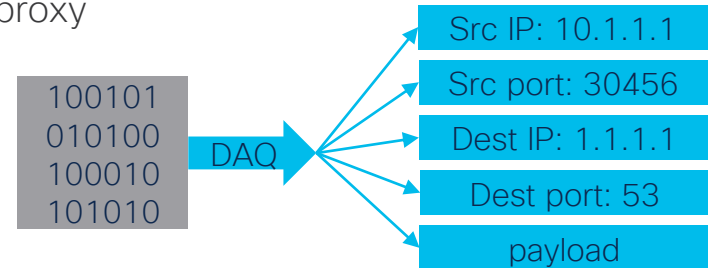
```
show asp inspect-dp snort queue-exhaustion
```

**Display automatic capture of PDTs ring when snort is unable to service queue**

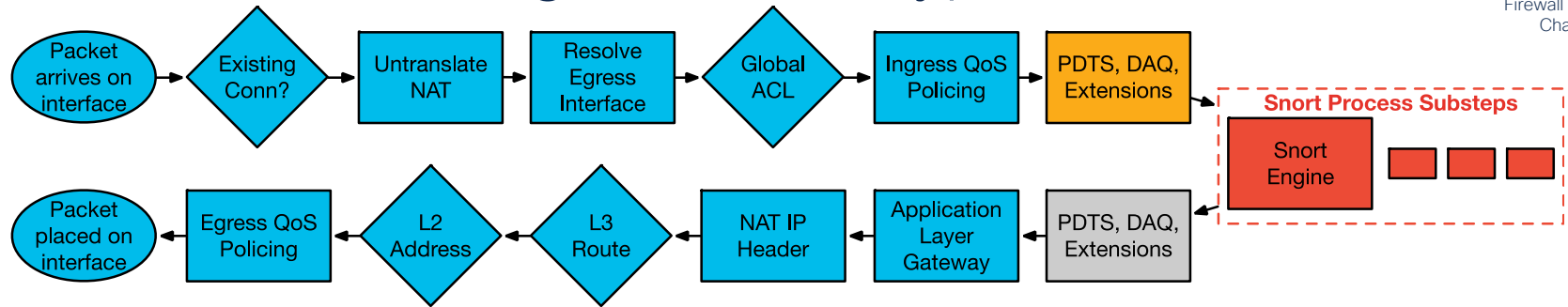
# Packet Processing: Data Acquisition Library



- The Data Acquisition Library (DAQ) translates packet data for snort so it can run on different hardware and software platforms
- Platform-specific changes are made in the DAQ
- DAQ extensions facilitate TLS decryption and a TCP proxy
- Decrypted flows are sent to snort for inspection
- Packets should not be dropped by the DAQ



# Packet Processing: SSL Decryption



- SSL Decryption touches Lina, DAQ, and Snort

## Lina and DAQ

- Proxy TCP sessions
- Track keys/sessions
- Decrypt (software) / send to crypto chip to decrypt

## Snort

- Enforces policies
- Makes decisions on whether to decrypt flow or not

# TLS Server Identity Discovery

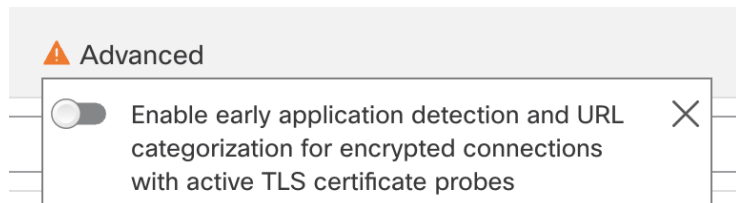


Cisco Secure  
Firewall YouTube  
Channel



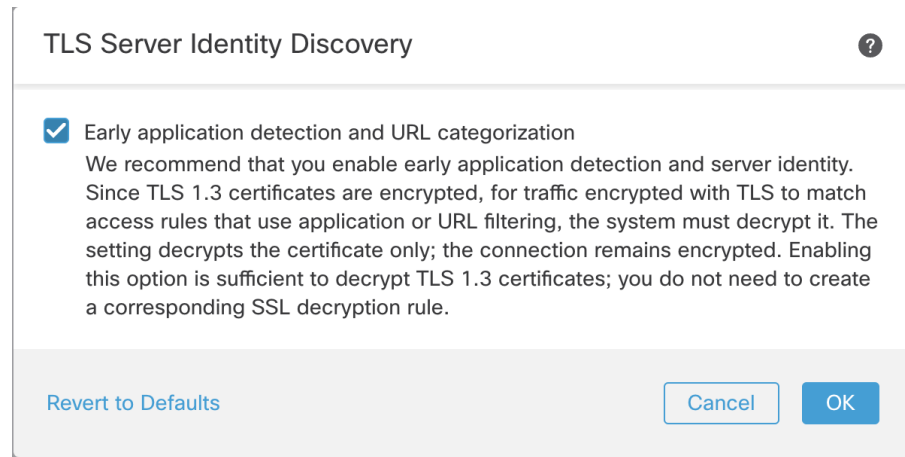
- Added in Version 6.7
- Enhances application and URL visibility for TLS 1.3 connections

## AC Policy > Advanced [TLS Server Identity Discovery]

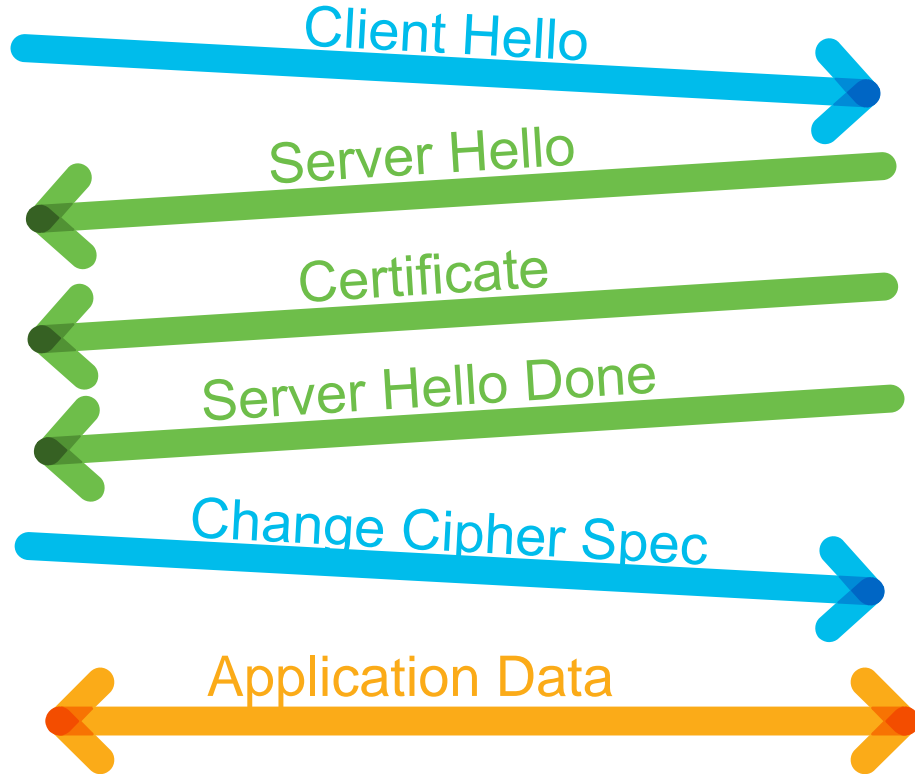


Blog post:

<https://blogs.cisco.com/security/network-security-efficacy-in-the-age-of-pervasive-tls-encryption>

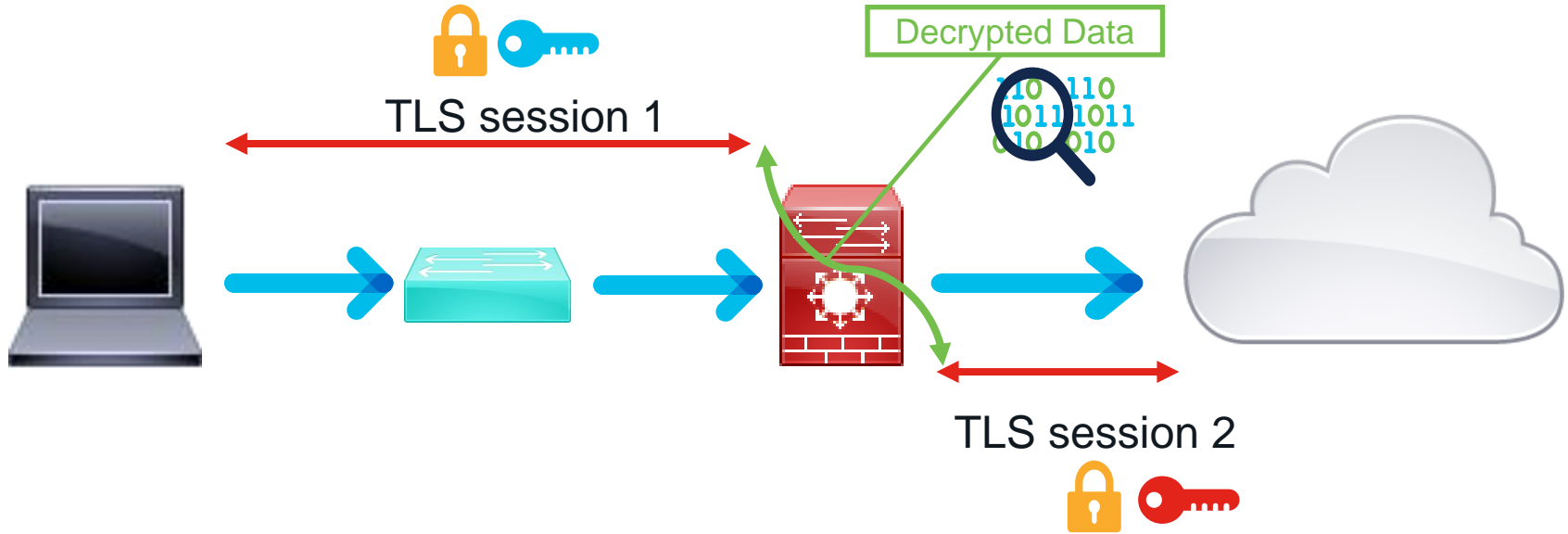


# Abbreviated SSL handshake

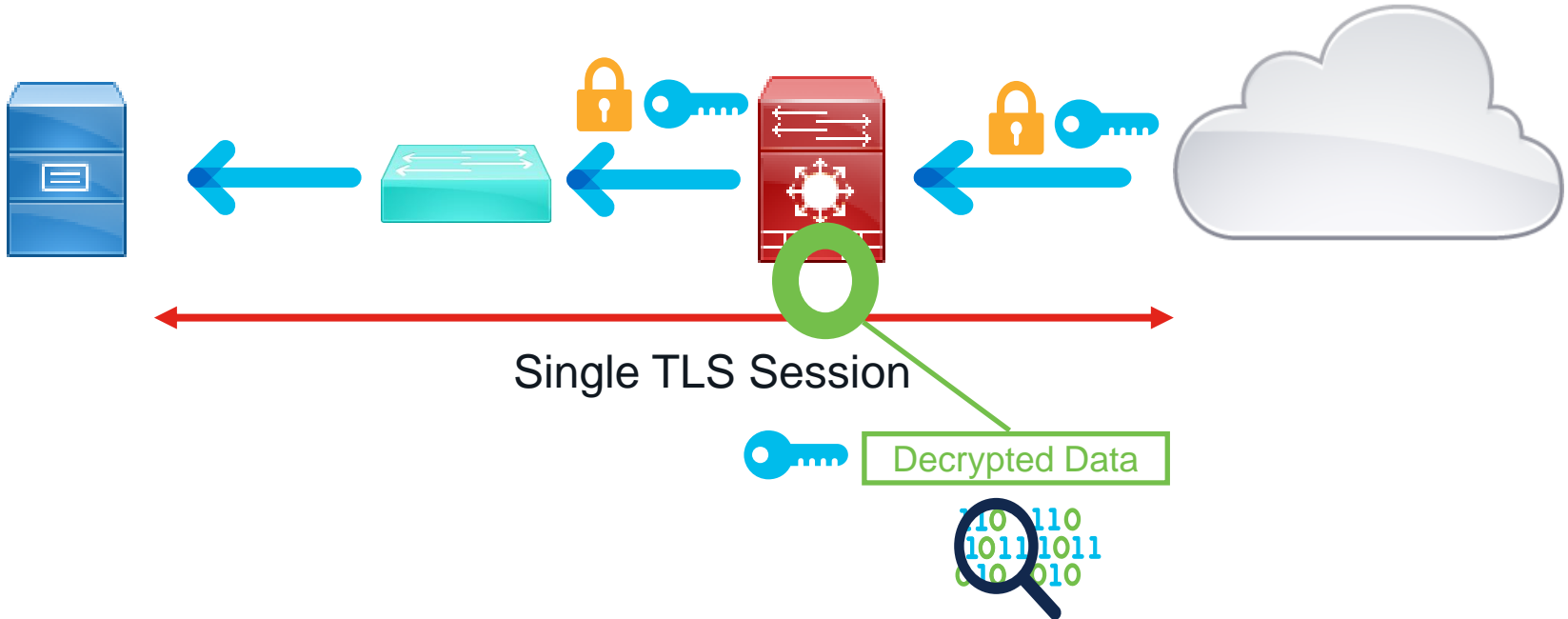




# Typical deployment: Decrypt Resign

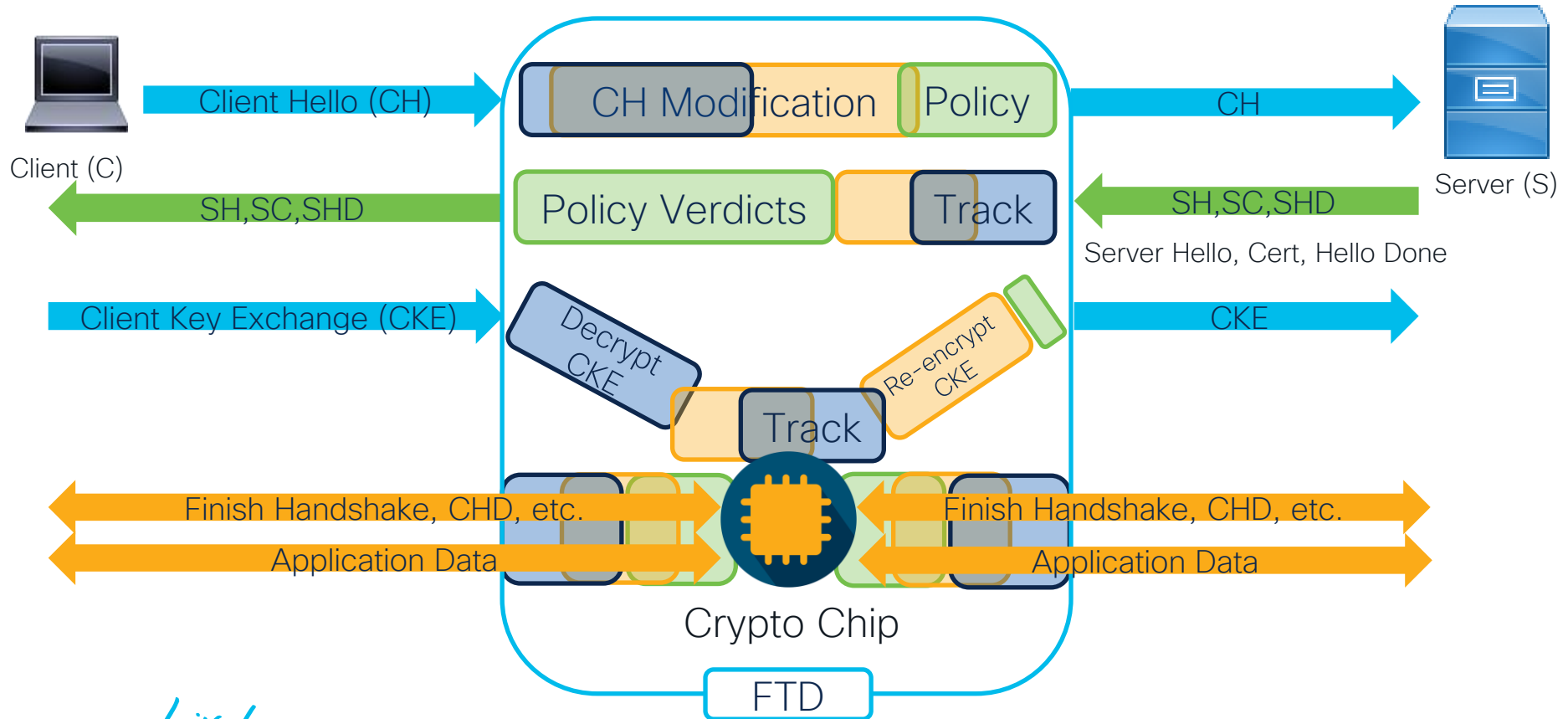


# Typical deployment: Decrypt Known-key

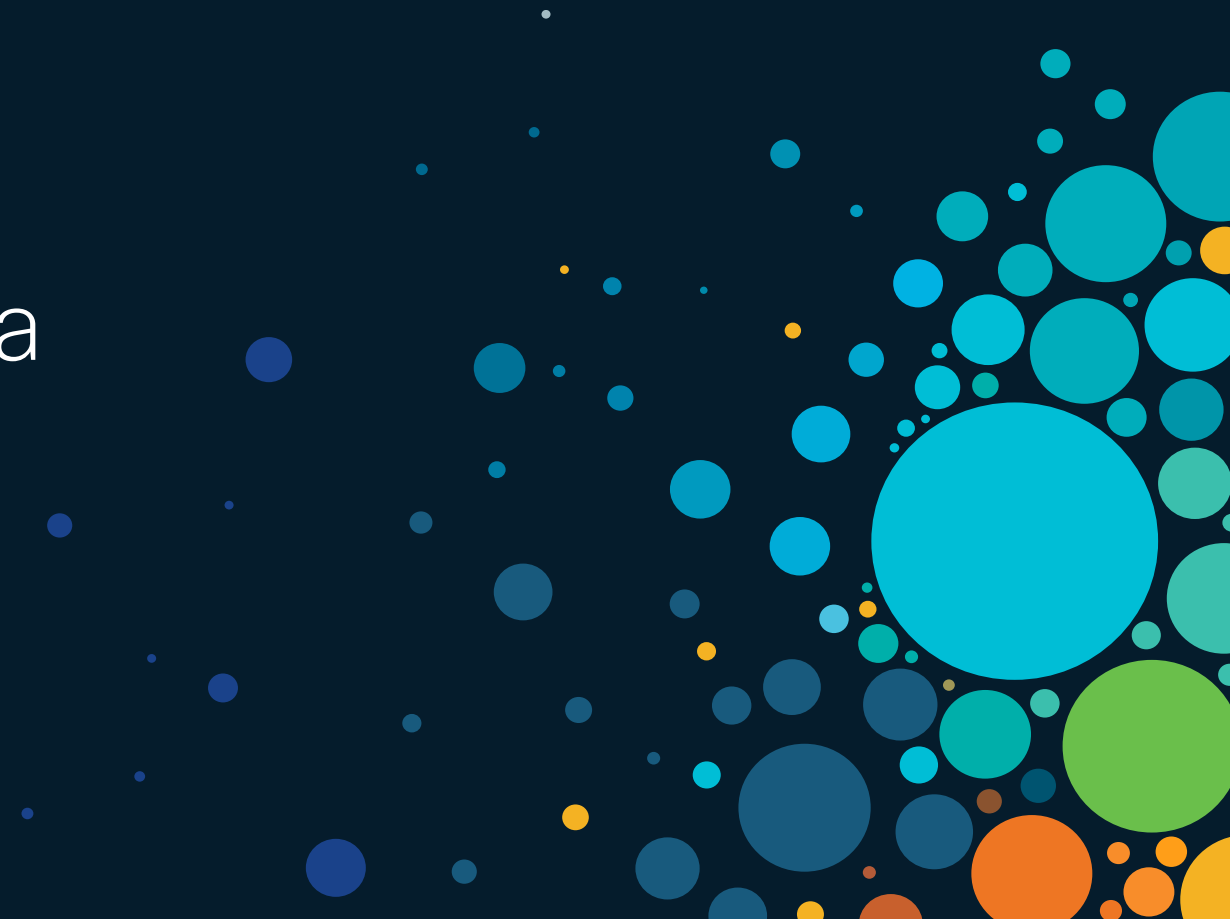


# Packet Processing: SSL Hardware Offload

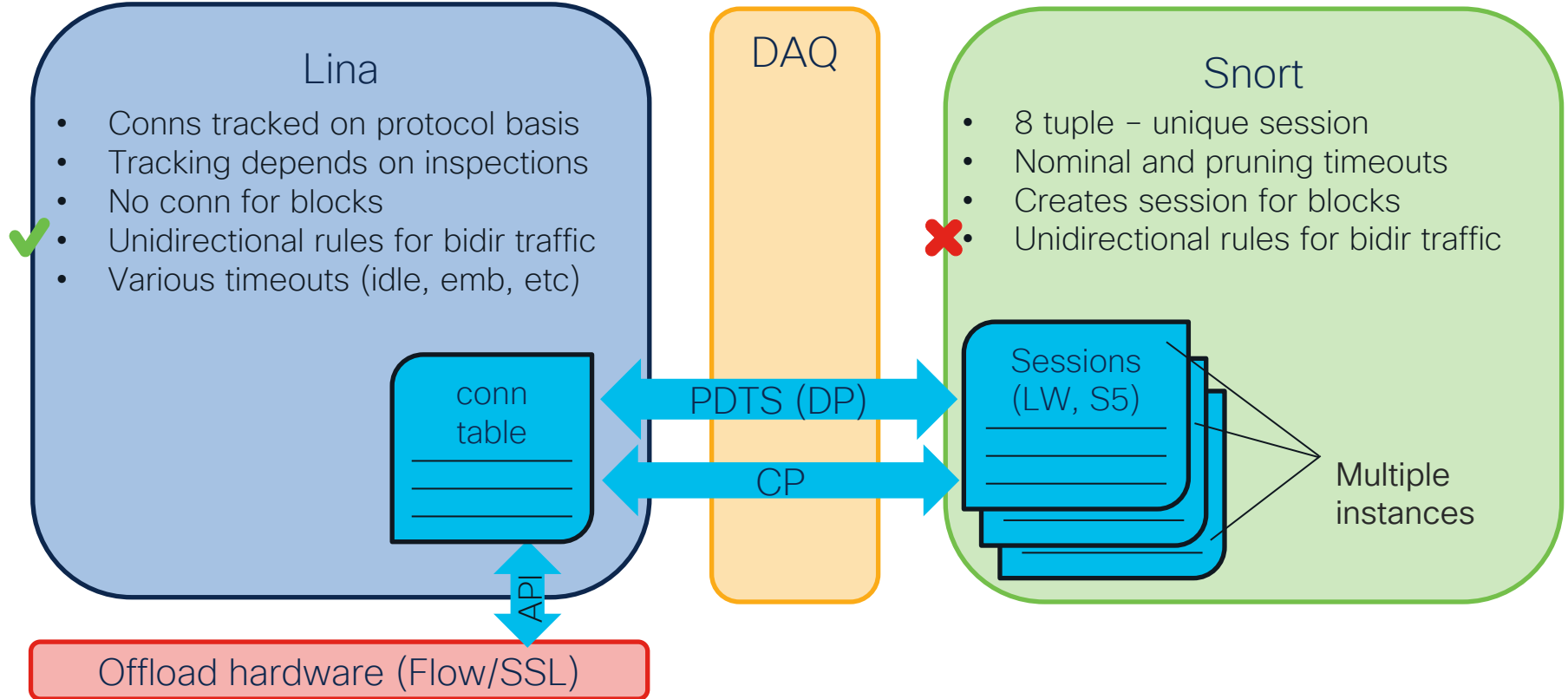
Lina DAQ Snort



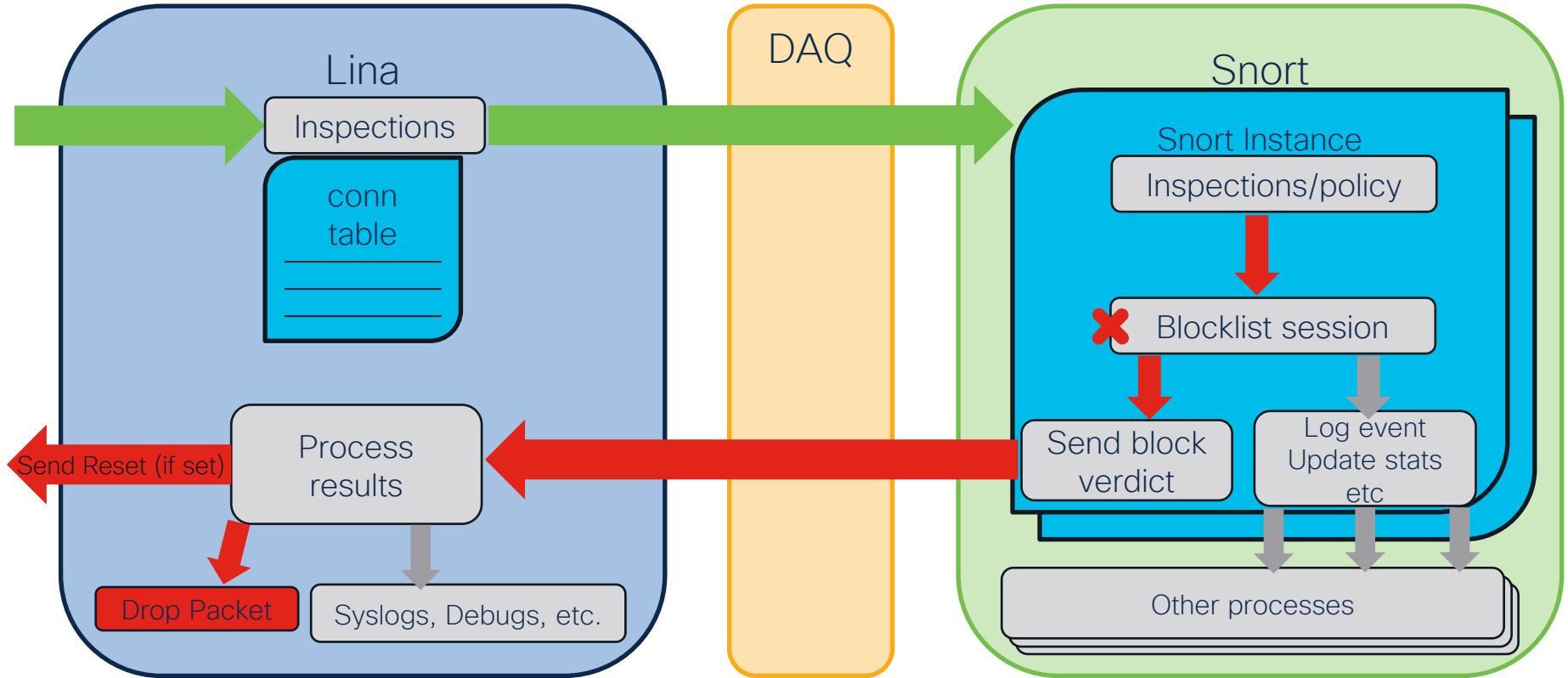
# Snort and Lina Interactions



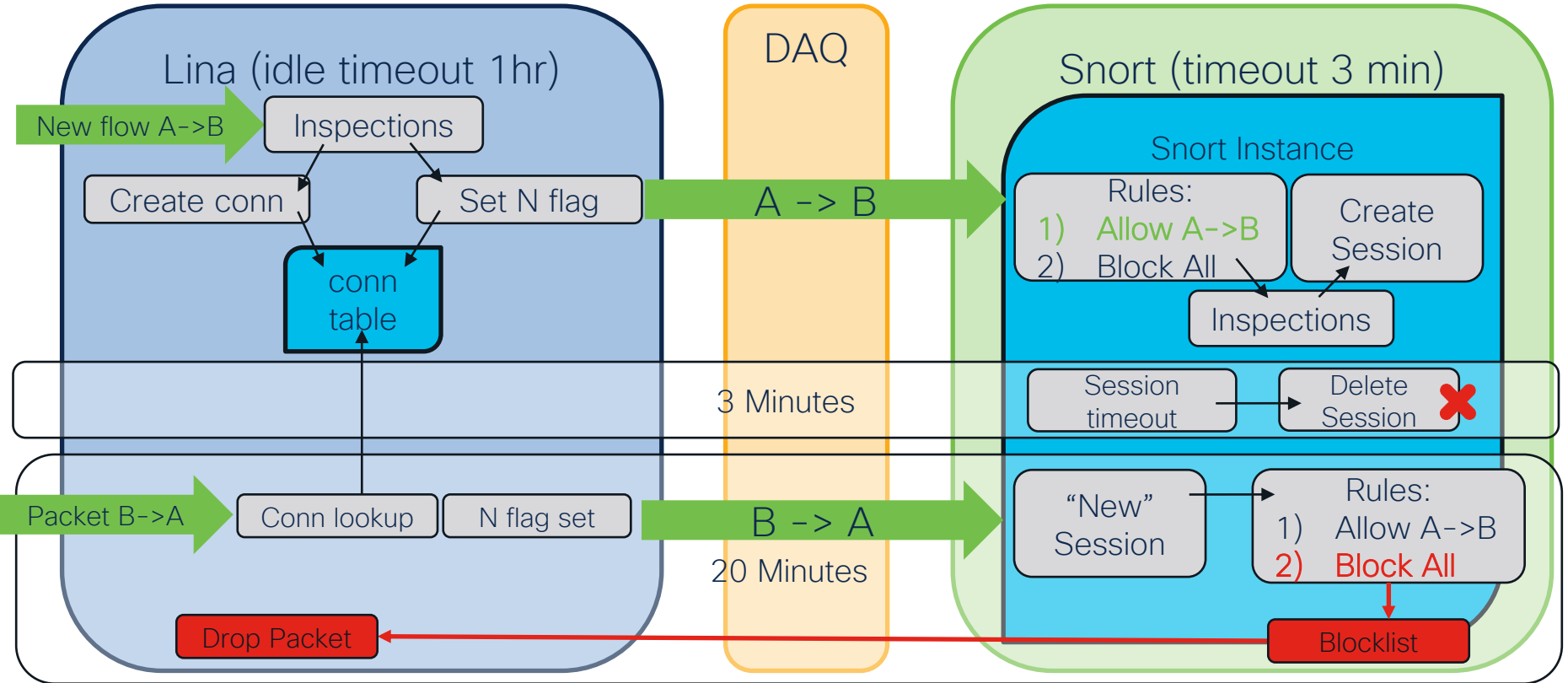
# Session Tracking



# Example flow – packet blocked by snort



# Example conn timeout (TCP) on version < 6.3



# Changes in 6.3+ for session tracking lina/snort (TCP Only)

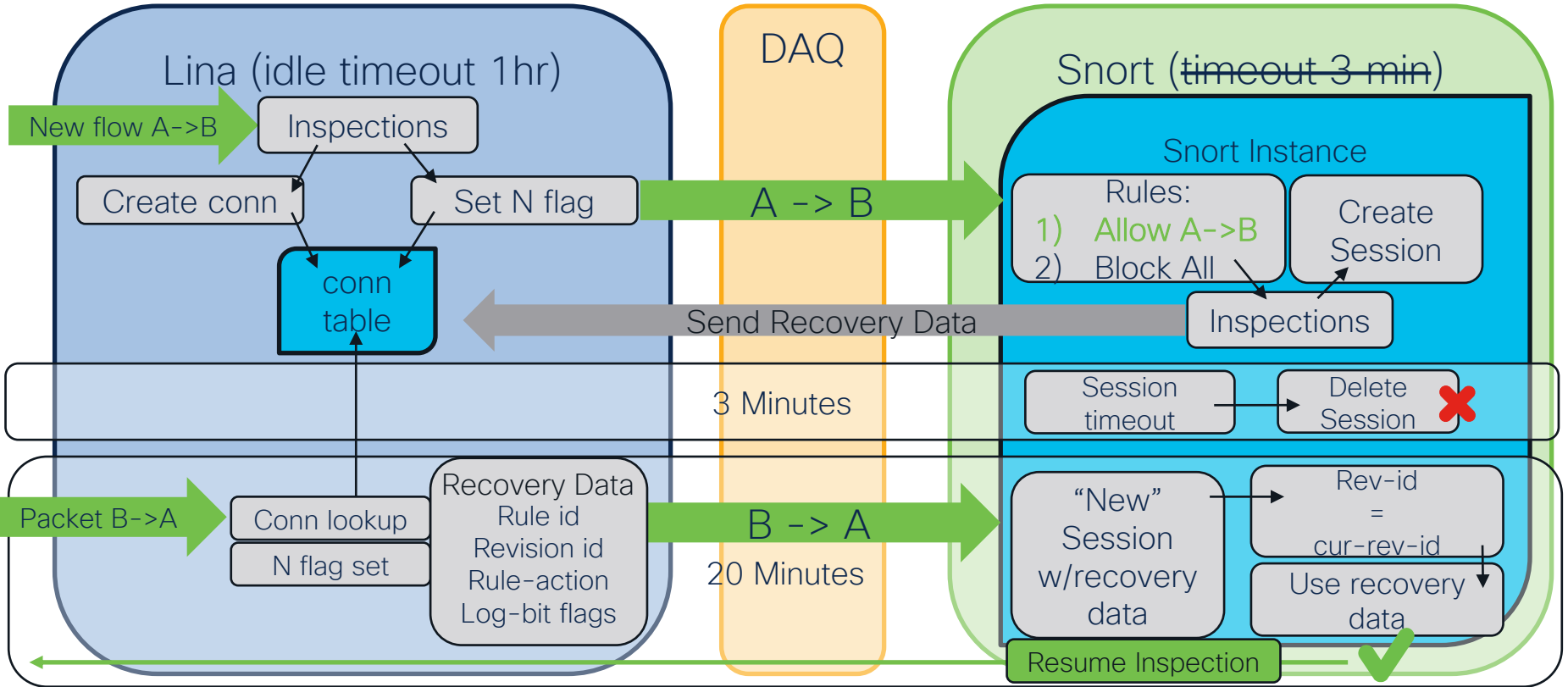


For your  
reference

- **Lina** sets timeouts and syncs them to **Snort**
- **Snort** sends **Lina** recovery data (RD) for each session
- **Lina** stores RD in conn-meta
- **Snort** queries **Lina** for RD if it doesn't know about a session
- Uses recovery data to match AC rule if revision hasn't changed
- When a conn times out in **Lina**, it sends **Snort** End of Flow (EOF) message



# Example conn timeout (TCP) on version 6.3+



# Configure timeouts in 6.3+ Threat Defense Service Policy

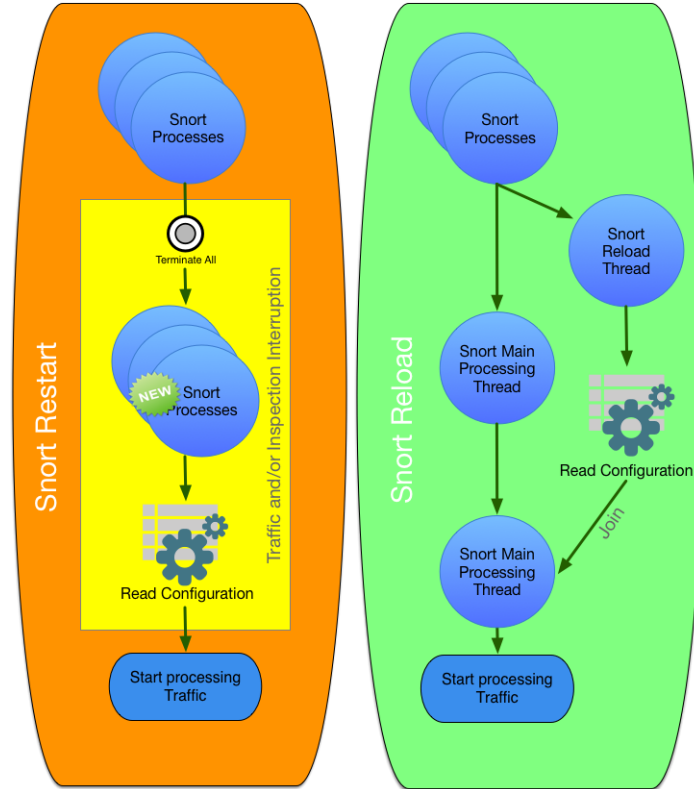
## AC Policy > Advanced

The screenshot displays the configuration page for a Threat Defense Service Policy. The left sidebar shows a navigation menu with categories: Prefilter Policy Settings, Network Analysis and Intrusion Policies, Threat Defense Service Policy (highlighted), and Files and Malware Settings. The main content area is titled 'Threat Defense Service Policy' and is divided into three numbered steps: 1. Interface Object, 2. Traffic Flow, and 3. Connection Setting (the current step). The 'Connection Setting' section includes the following options and fields:

- Enable TCP State Bypass
- Randomize TCP Sequence Number
- Enable Decrement TTL
- Connections:**
  - Maximum TCP & UDP:
  - Maximum Embryonic:
- Connections Per Client:**
  - Maximum TCP & UDP:
  - Maximum Embryonic:
- Connections Timeout:**
  - Embryonic:
  - Half Closed:
  - Idle:
- Reset Connection Upon Timeout
- Detect Dead Connections
  - Detection Timeout:
  - Detection Retries:

At the bottom right, there are navigation buttons: '<< Previous', 'Finish', and 'Cancel'.

# Snort Restart & Reload Architecture



# Why does Snort have to restart?

- New version of Snort in policy deploy
- Reallocate memory for pre-processors/Security Intelligence
- Reload shared objects
- Pre-processor configuration changes
- Configured to restart instead of reload

The screenshot shows the Cisco Firepower Management Center interface for editing a policy named "Snort Rule Test Policy". The "Advanced" tab is selected, and the "Inspect traffic during policy apply" option is checked and highlighted with a red box. Other settings include "Maximum URL characters to store in connection events" (1024), "Allow an Interactive Block to bypass blocking for (seconds)" (600), "Retry URL cache miss lookup" (Yes), "Enable Threat Intelligence Director" (Yes), "Enable reputation enforcement on DNS traffic" (Yes), "Identity Policy" (None), "SSL Policy to use for inspecting encrypted connections" (None), and "Early application detection and URL categorization" (Disabled).

Rules	Security Intelligence	HTTP Responses	Logging	Advanced
<b>General Settings</b>				
Maximum URL characters to store in connection events				1024
Allow an Interactive Block to bypass blocking for (seconds)				600
Retry URL cache miss lookup				Yes
Enable Threat Intelligence Director				Yes
Enable reputation enforcement on DNS traffic				Yes
Inspect traffic during policy apply				Yes
<b>Identity Policy Settings</b>				
Identity Policy				None
<b>SSL Policy Settings</b>				
SSL Policy to use for inspecting encrypted connections				None
<b>TLS Server Identity Discovery</b>				
Early application detection and URL categorization				Disabled

## Full listing of restart reasons

[https://www.cisco.com/c/en/us/td/docs/security/firepower/70/configuration/guide/fpmc-config-guide-v70/policy\\_management.html?bookSearch=true#concept\\_33516C5D6B574B6888B1A05F956ABDF9](https://www.cisco.com/c/en/us/td/docs/security/firepower/70/configuration/guide/fpmc-config-guide-v70/policy_management.html?bookSearch=true#concept_33516C5D6B574B6888B1A05F956ABDF9)

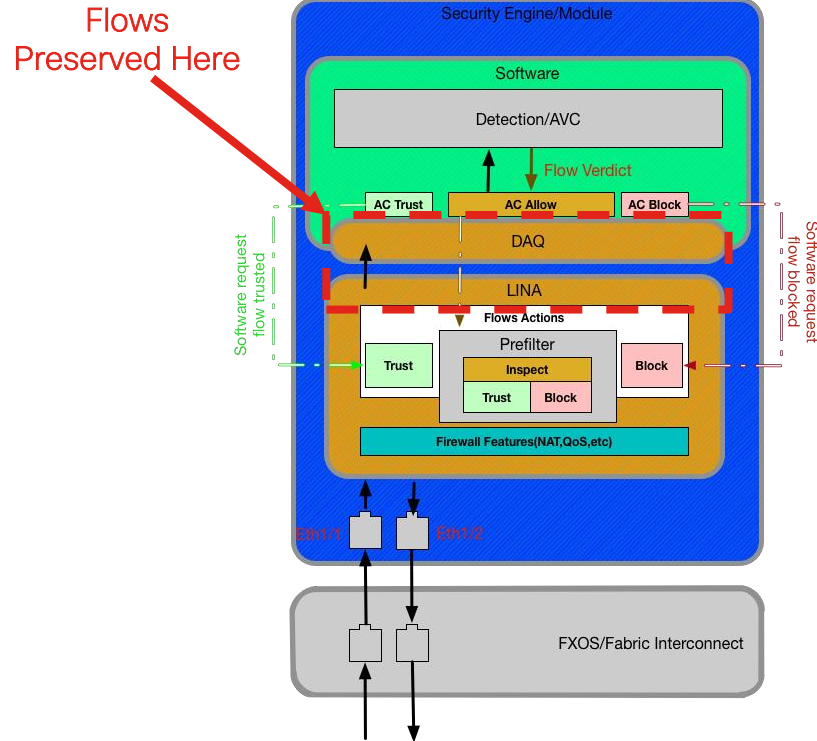
# Mitigations

1 Snort Preserve-Connection

2 Software Bridge

# Snort Preserve-Connection

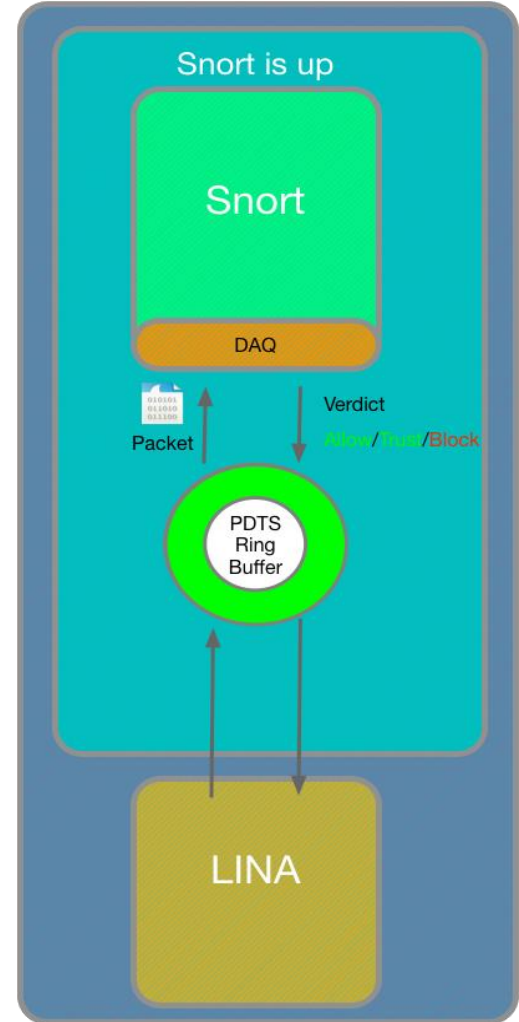
- When Snort goes down connections with Allow verdict are preserved in LINA
- Snort does **NOT** do a mid-session pickup on preserved flows on coming up
- Does **NOT** protect against new flows while Snort is down
- 6.2.3 Feature Introduction
- Can be enabled/disabled from CLISH: configure snort preserve-connection enable/disable



# Software Bypass

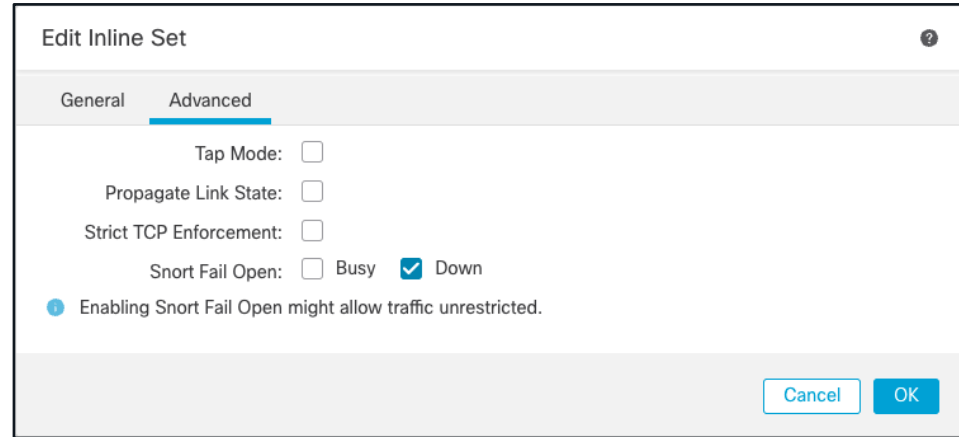
- With inline Fail-Open deployments traffic is passed uninspected on the Software bridge when Snort is down.
- When Snort comes up, Snort does a mid-session pickup on traffic
- A.K.A Software Bypass
- CLISH Command:

```
> pmtool disablebytype de
```



# Snort Fail-Open when Busy / Down

- Snort fail-open when down means that all traffic will pass over software bridge when snort is down
- Snort fail-open when busy means traffic will be bypassed around Snort when the incoming buffer for snort reaches 85% full



The screenshot shows the 'Edit Inline Set' configuration window with the 'Advanced' tab selected. The 'Snort Fail Open' option is checked for 'Down'.

Edit Inline Set

General Advanced

Tap Mode:

Propagate Link State:

Strict TCP Enforcement:

Snort Fail Open:  Busy  Down

Enabling Snort Fail Open might allow traffic unrestricted.

Cancel OK



# Packet Processing: Decode Preprocessor (GID:116)

Decode performs basic checks on packets like:

- Confirm Ethernet protocol matches IPv4 or IPv6 value
- Verify IPv4 header is at least 20 bytes

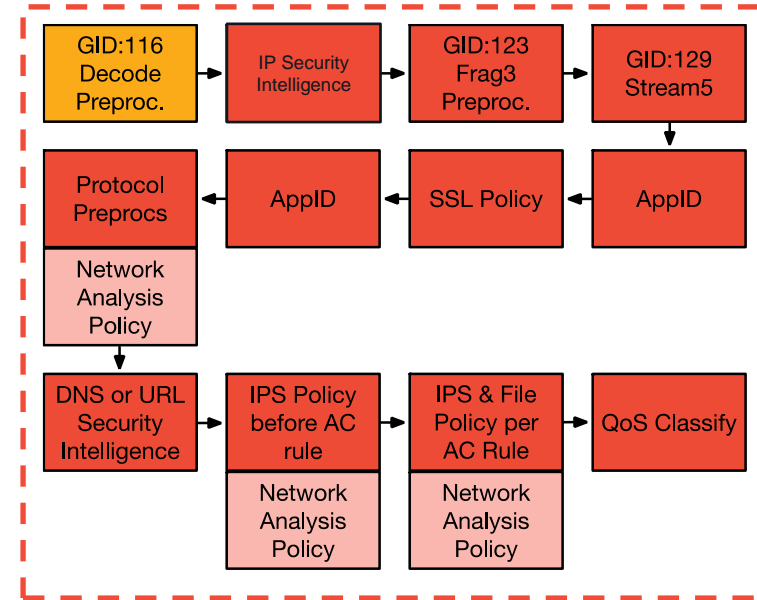
Very rare for Decode to produce unexpected packet drops

Set GID:116 rules to “generate events” for visibility

The screenshot shows the 'Rules' configuration window in Cisco Packet Tracer. The 'Filter' is set to 'GID:"116"'. Below the filter, it indicates '0 selected rules of 153'. A table lists the rules for GID 116:

Rule State	Event Filtering	Dynamic State	Alerting	Comments
<input type="checkbox"/>				
<input type="checkbox"/>				
<input type="checkbox"/>				
<input type="checkbox"/>				
<input type="checkbox"/>				
<input type="checkbox"/>				

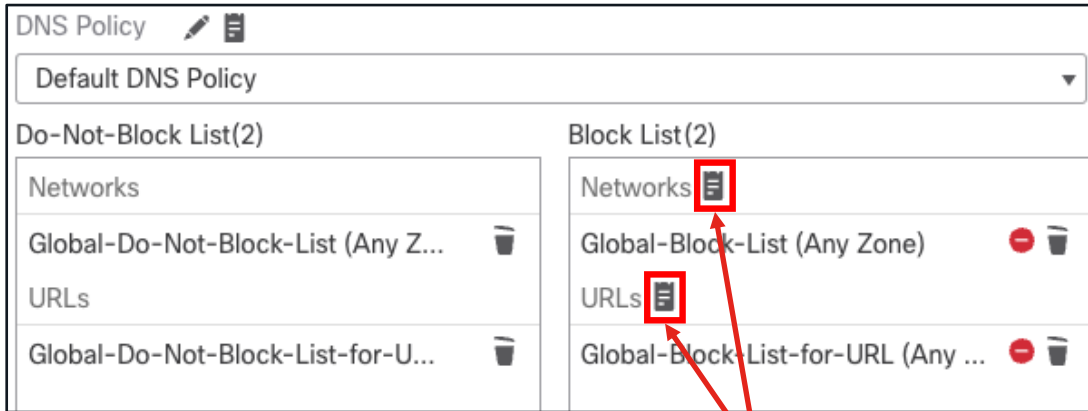
The table columns are: GID, SID, and Message. The messages listed are: DECODE\_NOT\_IPV4\_DGRAM, DECODE\_IPV4\_INVALID\_HEADER\_LEN, DECODE\_IPV4\_DGRAM\_LT\_IPHDR, DECODE\_IPV4OPT\_BADLEN, and DECODE\_IPV4OPT\_TRUNCATED.



Snort Process Substeps

# Packet Processing: IP Security Intelligence

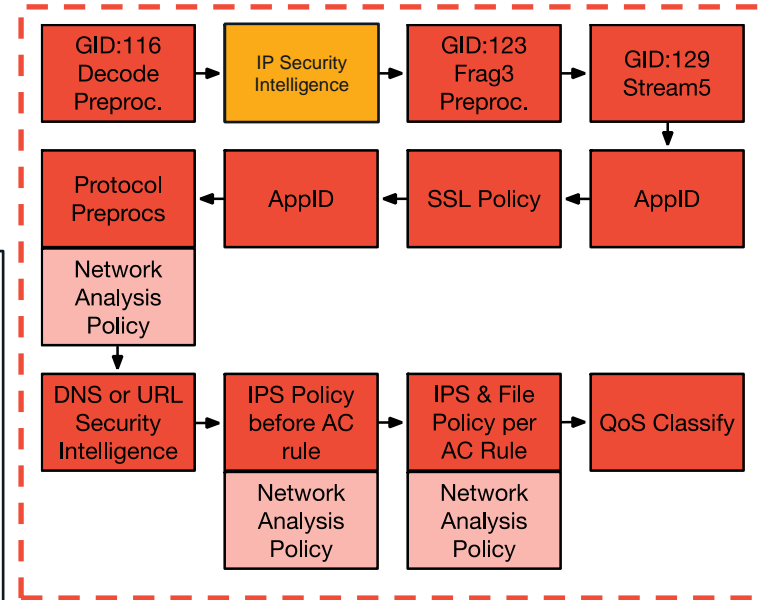
- IP SI drops packets based on lists of malicious IP addresses
- SI drops packets at the IP-level without higher layer inspects
- The **Do-Not-Block List** only overrides the **Block List**



Best Practice: **Log SI block-list events**

Verify an IP is on a block list:

```
$ grep -r [IP_ADDRESS] /var/sf/iprep_download
```



**Snort Process Substeps**

# Packet Processing: Frag Preprocessor (GID:123)

Frag3 reassembles IP fragments before higher-level preprocs

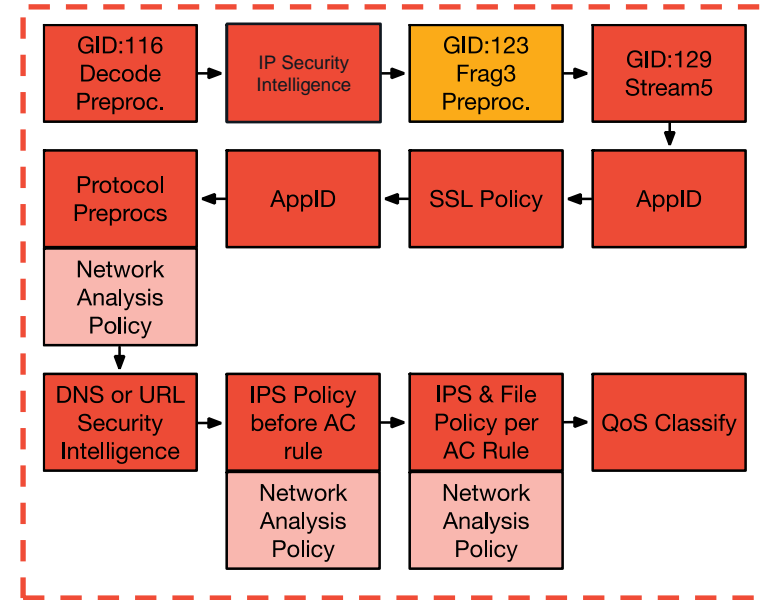
Rare, but possible causes for drops:

- Zero-byte fragments
- Overlapping fragments

Set GID:123 rules to “generate events” for visibility

The screenshot shows the Cisco configuration interface for rules. The left sidebar lists various rule configuration options like Message, SID, GID, Reference, Action, Protocol, Direction, Source IP, Destination IP, Source port, Destination port, Rule Overhead, and Metadata. The main area shows a filter for 'GID:"123"' and a table of 11 selected rules. The table has columns for Rule State, Event Filtering, Dynamic State, Alerting, and Comments. The visible rules are:

Rule State	Event Filtering	Dynamic State	Alerting	Comments
<input type="checkbox"/>	GID	SID ↑		Message
<input type="checkbox"/>	123	1		FRAG3_IPOPTIONS
<input type="checkbox"/>	123	2		FRAG3_TEARDROP
<input type="checkbox"/>	123	3		FRAG3_SHORT_FRAG
<input type="checkbox"/>	123	4		FRAG3_ANOMALY_OVERSIZE
<input type="checkbox"/>	123	5		FRAG3_ANOMALY_ZERO

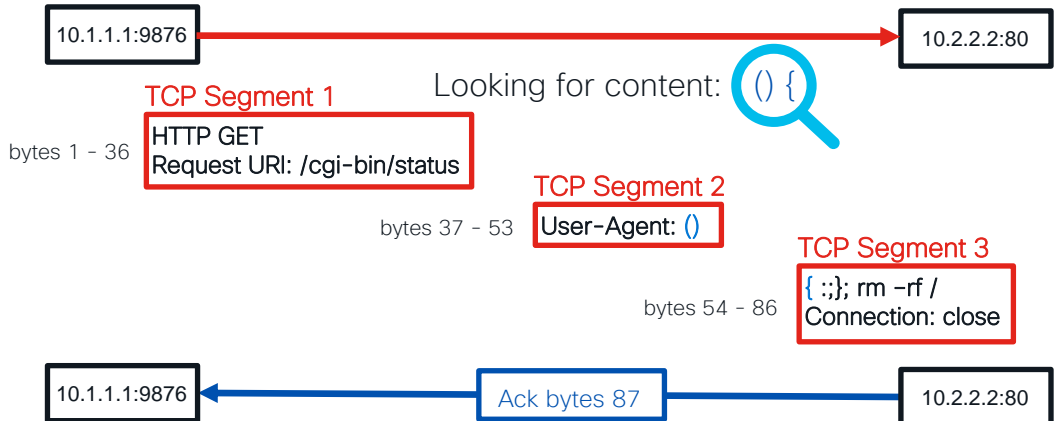


Snort Process Substeps

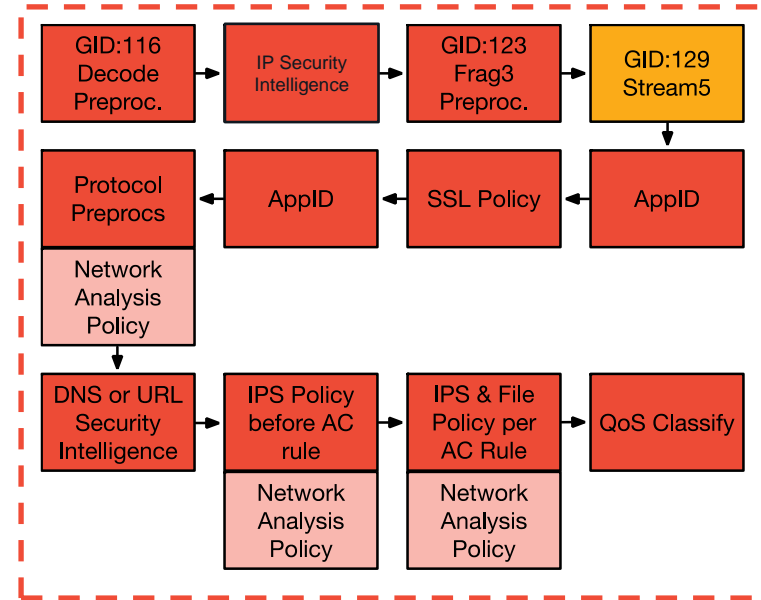
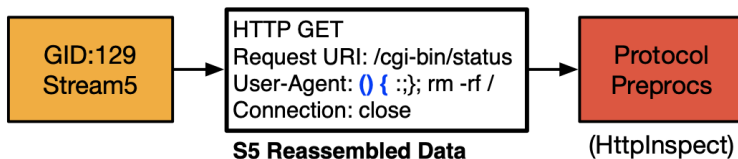
# Packet Processing: Stream Preproc (GID: 129)

- S5 Reassembles TCP segments for Protocol preprocs
- TCP segments must be contiguous and acknowledged

Stream 5 Input: TCP Segments



Stream Output: Data for Protocol Preprocs



Snort Process Substeps



# Stream5 Asymmetric Traffic Reference Slides

## Packet Processing: Stream5 Asymmetry Problem

- Snort sees half of the traffic for a given TCP session
- Snort receives TCP segments from 10.1.1.1 to 10.2.2.2, but not
- Segments stay in memory waiting for an ACK for reassembly, or
- Common causes: Portchannel interfaces which map to interface asymmetric routing where the sensor only sees one direction of

CiscoLive!

## Packet Processing: Stream5 Asymmetry Problem

S5 syslog messages observed in /var/log/messages:

```
S5: Session exceeded configured max bytes to queue xxxxx using xxxxx bytes
S5: Session exceeded configured max segs to queue xxxxx using xxxxx bytes
S5: Pruned session from cache that was using xxxxx bytes
S5: Pruned 5 sessions from cache for memcap. xxxxx ssnns remain
```

These syslogs may also be symptomatic of large TCP flows (such as backups), or snort instance oversubscription

CiscoLive!

## Reference Slide: Check Stream5 Asymmetry

Check for asymmetry by displaying TCP SYN to SYN-ACK ratio for all snort instances:

- > expert
- cd /var/sf/detection\_engines/[UUID]
- for i in \$(ls -lv | grep instance); do echo \$i; perfstats -q < \$i/now | grep -i "syns/sec" -A 1; done

Asymmetric output (BAD):

```
instance-1
Syns/Sec: 179.1 # ratio is far from 1:1
SynAcks/Sec: 2.3
```

Symmetric output (GOOD):

```
instance-1
Syns/Sec: 77.8 # ratio is almost 1:1
SynAcks/Sec: 79.1
```

Snort Process Substeps

CiscoLive!

# Asymmetric Traffic – TAC Script

Snort  
performance



## getS5HostInfo

- Script developed by TAC to get information about asymmetric traffic
- Available currently at: <https://github.com/johnjg12/snort-scripts>
- Generates CSV files and report files using syslog files (/var/log/messages)
- Hidden slides with details available in presentation PDF



# Story Time!

<input type="checkbox"/>	▼ <u>First Packet</u> ✕	<u>Last Packet</u> ✕	<u>Action</u> ✕	<u>Initiator IP</u> ✕	<u>Responder IP</u> ✕	<u>Ingress Security Zone</u> ✕	<u>Source Port / ICMP Type</u> ✕	<u>Destination Port / ICMP Code</u> ✕
↓ <input type="checkbox"/>	<a href="#">2018-06-05 19:33:32</a>	<a href="#">2018-06-05 19:35:33</a>	Allow	 192.168.0.4	 8.8.8.8	Passive	12755 / udp	53 (domain) / udp
↓ <input type="checkbox"/>	<a href="#">2018-06-05 19:33:32</a>	<a href="#">2018-06-05 19:35:33</a>	Allow	 8.8.8.8	 192.168.0.4	Passive	53 (domain) / udp	12755 / udp
↓ <input type="checkbox"/>	<a href="#">2018-06-05 19:33:32</a>	<a href="#">2018-06-05 19:35:33</a>	Allow	 8.8.8.8	 192.168.0.4	Passive	53 (domain) / udp	12434 / udp
↓ <input type="checkbox"/>	<a href="#">2018-06-05 19:33:32</a>	<a href="#">2018-06-05 19:35:33</a>	Allow	 192.168.0.4	 8.8.8.8	Passive	12434 / udp	53 (domain) / udp

⏪ Page 1 of 1 ⏩ Displaying rows 1-4 of 4 rows

```
> show version
```

```
-----[ Cartographer ]-----  
Model          : Cisco ASA5506-X Threat Defense (75) Version 6.2.3.1 (Build 43)  
UUID           : 8bd92a22-b2c1-11e7-a279-d47df0c19fbd  
Rules update version : 2018-05-30-001-vrt  
VDB version     : 297  
-----  
T
```

```
> █
```



# Asymmetric Traffic – Common Problems

Configuration options



Problem:

Different VLANs on each side of session

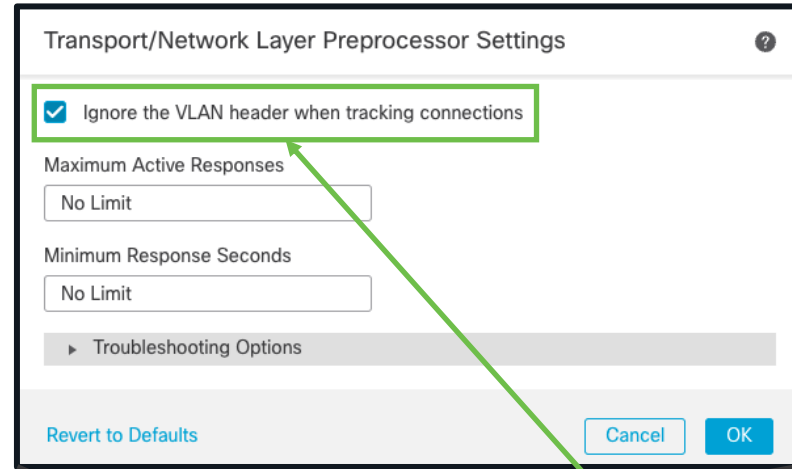
Example:

(VLAN50) 192.168.1.2 -> 10.8.0.2

(VLAN51) 10.8.0.2 -> 192.168.1.2

Fix:

Enable VLAN agnostic mode



Check this box to ignore VLANs when identifying unique sessions

Access Control Policy Advanced tab

Transport/Network Layer Preprocessor Settings

Ignore the VLAN header when tracking connections

No

# Asymmetric Traffic – Common Problems

Configuration options



Problem:

Traffic from same session traversing multiple Inline sets

Example:

Inline set A 192.168.1.2 -> 10.8.0.2

Inline set B 10.8.0.2 -> 192.168.1.2

Fix:

Combine pairs into single inline set

Devices > Device Management [Edit device]

Separate inline sets

Device	Interfaces	Inline Sets
<b>Name</b>		
<b>Interface Pairs</b>		
Inline Set A	s1p1 ↔ s1p2	
Inline Set B	s1p3 ↔ s1p4	



Single inline set

Device	Interfaces	Inline Sets
<b>Name</b>		
<b>Interface Pairs</b>		
Single Inline Set	s1p1 ↔ s1p2, s1p3 ↔ s1p4	

# Asymmetric Traffic – Common Problems

Configuration options



Problem:

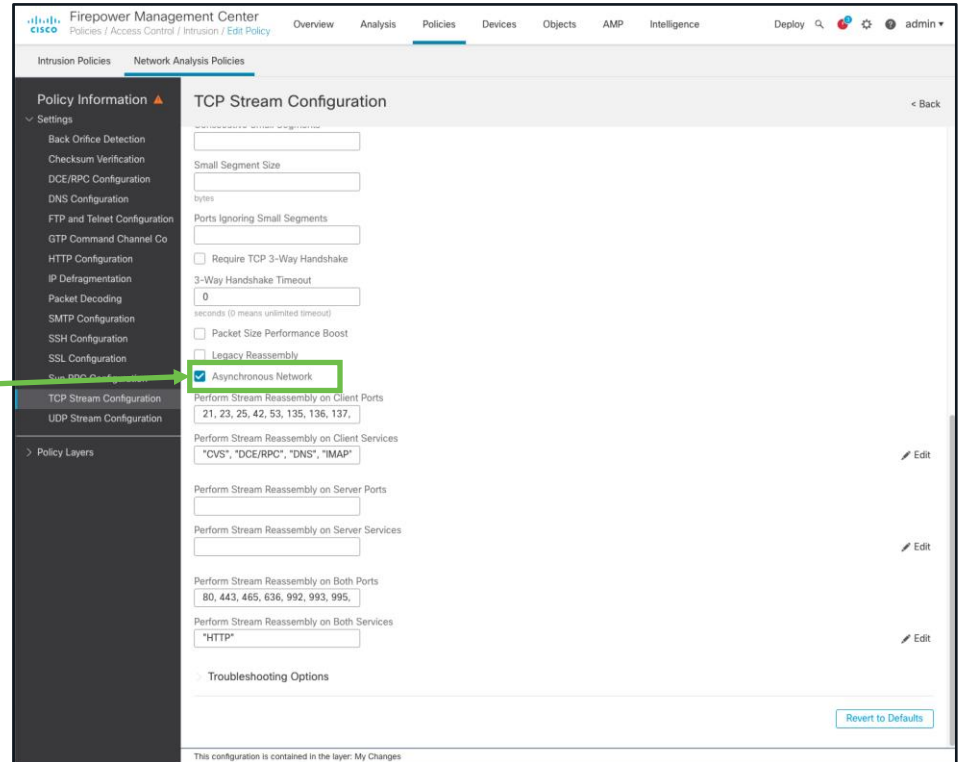
Traffic is actually asymmetric

Fix:

Configure network or move device so that there is no asymmetric traffic

Mitigation:

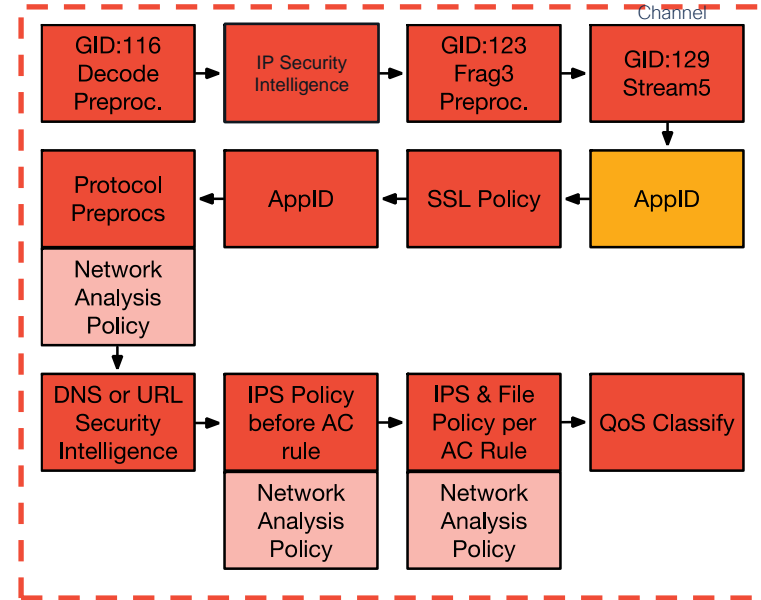
Enable Asynchronous Network option in NAP > TCP Stream Configuration



The screenshot shows the Firepower Management Center interface for configuring Network Analysis Policies (NAP). The left sidebar lists various settings, with 'TCP Stream Configuration' selected. The main panel displays the configuration for 'TCP Stream Configuration'. The 'Asynchronous Network' checkbox is checked and highlighted with a green box. A green arrow points from the text 'Enable Asynchronous Network option in NAP > TCP Stream Configuration' to this checkbox. Other settings include 'Small Segment Size', 'Ports Ignoring Small Segments', 'Require TCP 3-Way Handshake', '3-Way Handshake Timeout', 'Packet Size Performance Boost', 'Legacy Reassembly', 'Perform Stream Reassembly on Client Ports', 'Perform Stream Reassembly on Client Services', 'Perform Stream Reassembly on Server Ports', 'Perform Stream Reassembly on Server Services', 'Perform Stream Reassembly on Both Ports', and 'Perform Stream Reassembly on Both Services'. A 'Revert to Defaults' button is visible at the bottom right.

# Packet Processing: AppID

- AppID identifies over 3,500 layer 7 network applications:
  - Sharepoint, Facebook, Facebook chat, etc
- AppID runs both before and after SSL decryption
- AppID does not drop traffic
- An incorrect AppID disposition can cause traffic to match the wrong access control rule



Snort Process Substeps

Secure Firewall Application Detectors

Risk	Business Relevance	Tags	Categories
Very Low	1,407 Very High	310	active directory 8
Low	818 High	685	adult content 37
Medium	966 Medium	1,307	ad partial 182
High	276 Low	619	anonymizer/privacy 41
Very High	107 Very Low	813	allows remote connect 88
			allows remote control 51
			antivirus 13
			business 239

Application Details (3,704)

Application Name	Description	Risk	Business Relevance
050uk	VSP smartphone app.	Medium	Medium
181 Internet	Internet and Domain name service provider.	Very Low	Low
1-800-Flowers	Online retailer of flowers and other gifts.	Low	Very Low
1000minds	Advertising and analytics site.	Low	Very Low
100Bao	A Chinese P2P file sharing program.	Very High	Very Low
12306.cn	China Railway online customer service.	Very Low	High
123Movies	Online movie site.	Medium	Very Low

AppID Self-Help Portal



<https://appid.cisco.com>



# Packet Processing: AppID Debugging

- Specify flow 5-tuple of a flow to see application matching:

```
> system support application-identification-debug
```

Output:

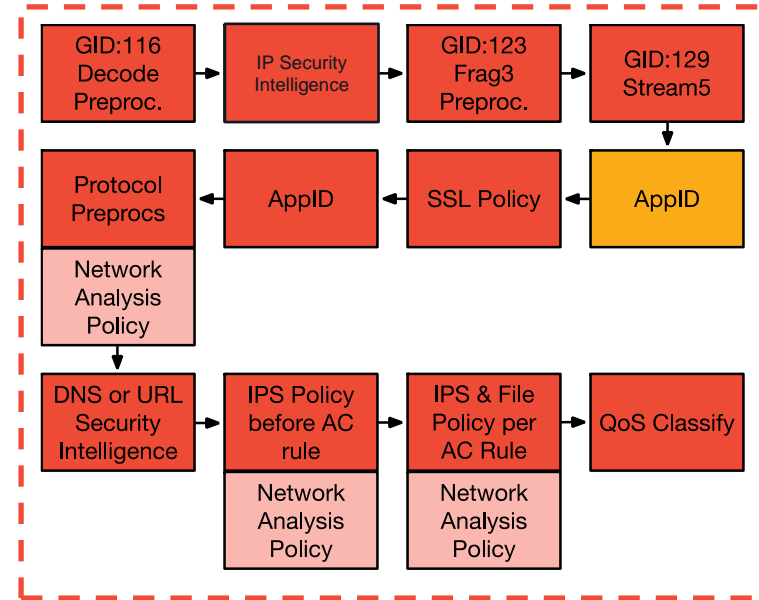
AS address space

I snort instance number

R 1<sup>st</sup> packet reversed (server to client)

- Specify flow 5-tuple to show access control rule matching:

```
> system support firewall-engine-debug
```



Snort Process Substeps

# Packet Processing: SSL Policy

An authorized man-in-the-middle of TLS/SSL traffic

For servers you own (Inbound traffic):

**Decrypt: Known Key** - Requires private key and certificate

For clients navigating to 3rd party sites (Outbound traffic):

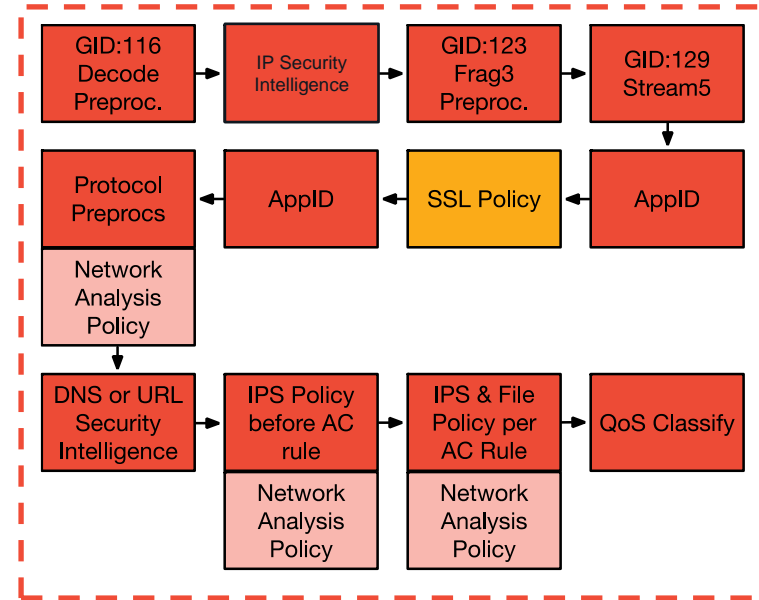
**Decrypt: Resign** - Resign certificate with an intermediate CA

Two options for new certificates to be trusted:

1. The client must trust the **FMC** as a **Certificate Authority**
2. The client must trust a CA which signs the FMC's CSR (Certificate Signing Request)

Traffic is TCP (SSL/TLS) and proxied in a DAQ extension which sends cleartext traffic to snort for IPS inspection.

**Note:** **TLS Server Identity Discovery** invokes SSL Policy features for TLS1.3 connections



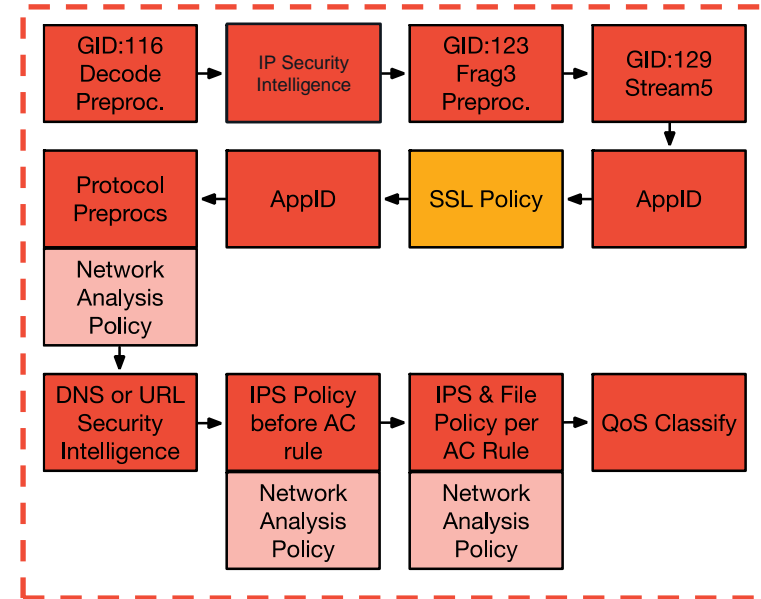
**Snort Process Substeps**

# Packet Processing: SSL Policy Debugging

Be careful with “**Undecryptable Actions**”, especially if your default action in the SSL Policy rules is “**Block**”

Rules	Trusted CA Certificates	Undecryptable Actions
Decryption Errors		Block
Handshake Errors		Inherit Default Action
Session not cached		Inherit Default Action
Unsupported Cipher Suite		Inherit Default Action
Unknown Cipher Suite		Inherit Default Action
SSLv2 Session		Inherit Default Action
Compressed Session		Inherit Default Action

[Revert to Defaults](#)



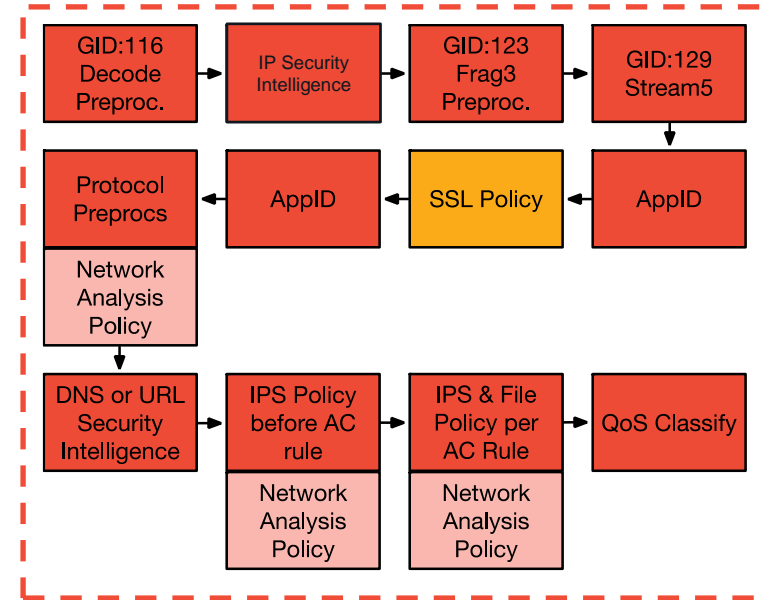
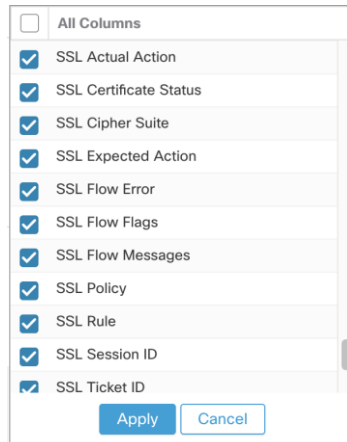
Snort Process Substeps

# Packet Processing: SSL Policy Debugging

Troubleshooting Best Practices:

- 1) Take note of browser side errors!
- 2) View SSL columns in Connection Events:
  - Navigate to “Analysis > Connections > Events”
  - Click “Table View of Connection Events”
  - Click “X” next to any column and select SSL columns


3) Columns in connection events explain decryption errors



Snort Process Substeps



# Connection Event Review



## Connection Events (switch workflow)

Connections with Application Details > [Table View of Connection Events](#)

Search Constraints ([Edit Search](#) [Save Search](#))

Jump to... ▾

	First Packet	Last Packet	Action	Reason	Initiator IP	Initiator Country	Responder IP	Responder Country
↓	2017-05-30 13:09:23	2017-05-30 13:09:24	Block	SSL Block	192.168.1.200		216.58.217.138	USA
↓	2017-05-30 13:08:53	2017-05-30 13:08:54	Block	SSL Block	192.168.1.200		216.58.217.138	USA
↓	2017-05-30 13:08:23	2017-05-30 13:08:24	Block	SSL Block	192.168.1.200		216.58.217.138	USA
↓	2017-05-30 13:08:19	2017-05-30 13:08:20	Block	SSL Block	192.168.1.200		216.58.217.138	USA
↓	2017-05-30 13:07:53	2017-05-30 13:07:54	Block	SSL Block	192.168.1.200		216.58.217.138	USA
↓	2017-05-30 13:07:23	2017-05-30 13:07:24	Block	SSL Block	192.168.1.200		216.58.217.138	USA

SSL Blocking flow

Cause of the SSL failure

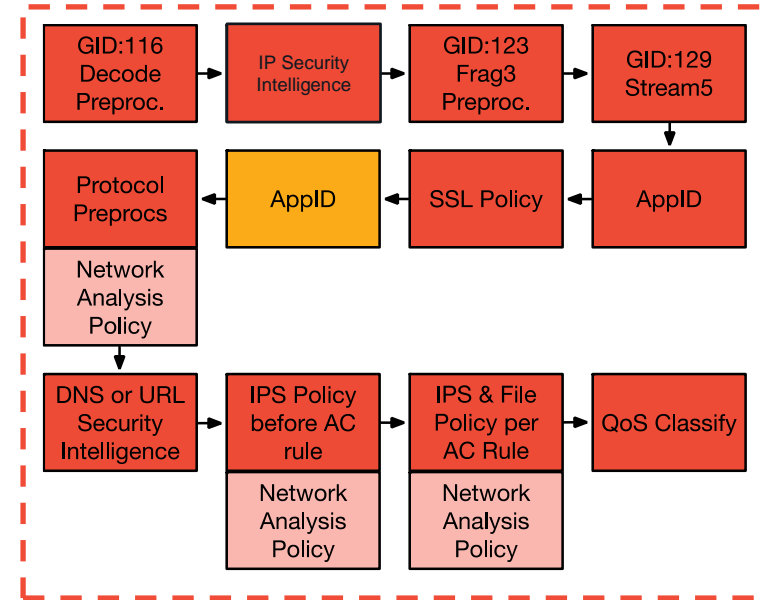
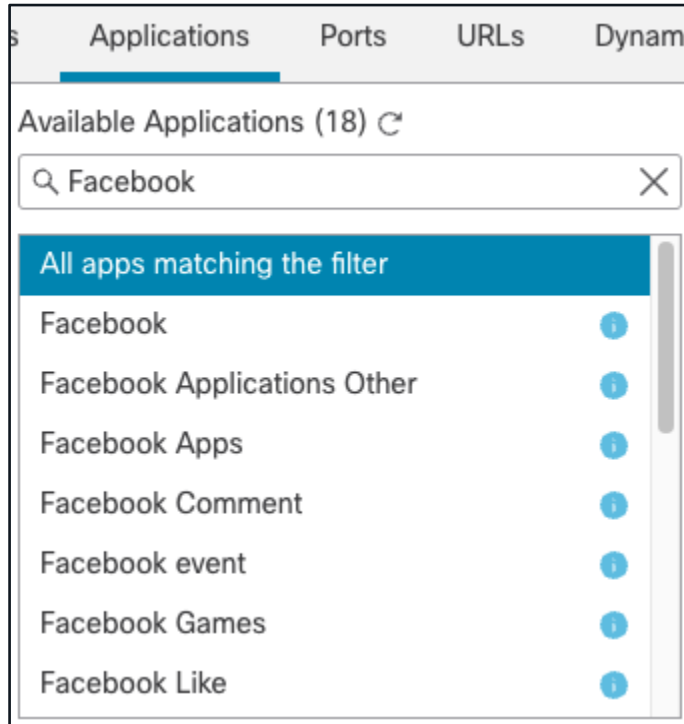
SSL flow flags for what happened with flow

SSL Status	SSL Flow Error	SSL Actual Action	SSL Expected Action	SSL Certificate Status	SSL Version
Decrypt (Resign)	PUB_CRYPTOPENSSL_RSA_OP_FAILURE (0xb7000a20)	Decrypt (Resign)	Decrypt (Resign)	Valid	TLSv1.2
Decrypt (Resign)	PUB_CRYPTOPENSSL_RSA_OP_FAILURE (0xb7000a20)	Decrypt (Resign)	Decrypt (Resign)	Valid	TLSv1.2
Decrypt (Resign)	PUB_CRYPTOPENSSL_RSA_OP_FAILURE (0xb7000a20)	Decrypt (Resign)	Decrypt (Resign)	Valid	TLSv1.2
Decrypt (Resign)	PUB_CRYPTOPENSSL_RSA_OP_FAILURE (0xb7000a20)	Decrypt (Resign)	Decrypt (Resign)	Valid	TLSv1.2
Decrypt (Resign)	PUB_CRYPTOPENSSL_RSA_OP_FAILURE (0xb7000a20)	Decrypt (Resign)	Decrypt (Resign)	Valid	TLSv1.2
Decrypt (Resign)	PUB_CRYPTOPENSSL_RSA_OP_FAILURE (0xb7000a20)	Decrypt (Resign)	Decrypt (Resign)	Valid	TLSv1.2

SSL Rule	SSL Session ID	SSL Ticket ID	SSL Flow Flags	SSL Flow Messages
MITM	0x0	0x0	VALID, INITIALIZED, SSL_DETECTED, CERTIFICATE_DECODED, FULL_HANDSHAKE, CLIENT_HELLO_SESSTKT, SERVER_HELLO_SESSTKT, CH_PROCESSED, SH_PROCESSED, CH_CIPHERS_MODIFIED, ...	CLIENT_HELLO, SERVER_HELLO, SERVER_CERTIFICATE
MITM	0x0	0x0	VALID, INITIALIZED, SSL_DETECTED, CERTIFICATE_DECODED, FULL_HANDSHAKE, CLIENT_HELLO_SESSTKT, SERVER_HELLO_SESSTKT, CH_PROCESSED, SH_PROCESSED, CH_CIPHERS_MODIFIED, ...	CLIENT_HELLO, SERVER_HELLO, SERVER_CERTIFICATE
MITM	0x0	0x0	VALID, INITIALIZED, SSL_DETECTED, CERTIFICATE_DECODED, FULL_HANDSHAKE, CLIENT_HELLO_SESSTKT, SERVER_HELLO_SESSTKT, CH_PROCESSED, SH_PROCESSED, CH_CIPHERS_MODIFIED, ...	CLIENT_HELLO, SERVER_HELLO, SERVER_CERTIFICATE
MITM	0x0	0x0	VALID, INITIALIZED, SSL_DETECTED, CERTIFICATE_DECODED, FULL_HANDSHAKE, CLIENT_HELLO_SESSTKT, SERVER_HELLO_SESSTKT, CH_PROCESSED, SH_PROCESSED, CH_CIPHERS_MODIFIED, ...	CLIENT_HELLO, SERVER_HELLO, SERVER_CERTIFICATE
MITM	0x0	0x0	VALID, INITIALIZED, SSL_DETECTED, CERTIFICATE_DECODED, FULL_HANDSHAKE, CLIENT_HELLO_SESSTKT, SERVER_HELLO_SESSTKT, CH_PROCESSED, SH_PROCESSED, CH_CIPHERS_MODIFIED, ...	CLIENT_HELLO, SERVER_HELLO, SERVER_CERTIFICATE
MITM	0x0	0x0	VALID, INITIALIZED, SSL_DETECTED, CERTIFICATE_DECODED, FULL_HANDSHAKE, CLIENT_HELLO_SESSTKT, SERVER_HELLO_SESSTKT, CH_PROCESSED, SH_PROCESSED, CH_CIPHERS_MODIFIED, ...	CLIENT_HELLO, SERVER_HELLO, SERVER_CERTIFICATE

# Packet Processing: AppID (Post SSL Decryption)

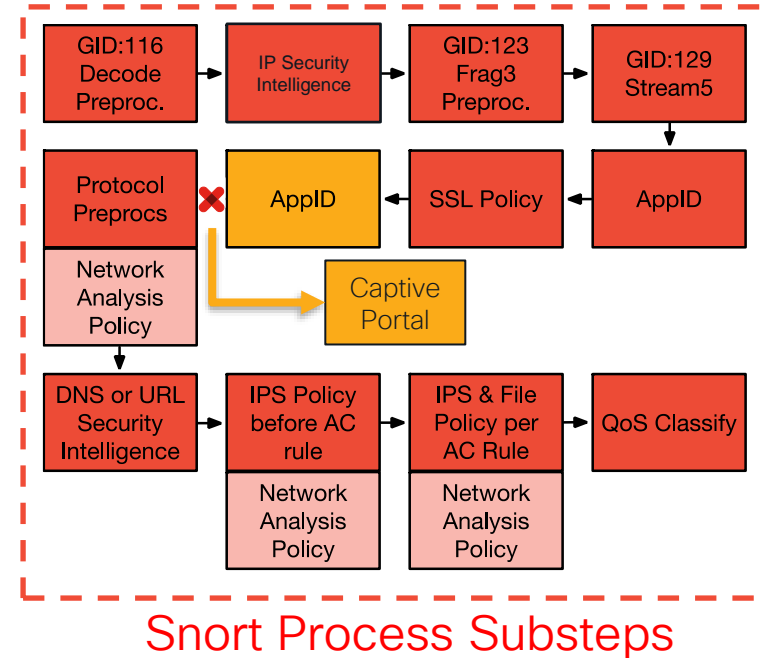
- Some apps require SSL decryption for further differentiation



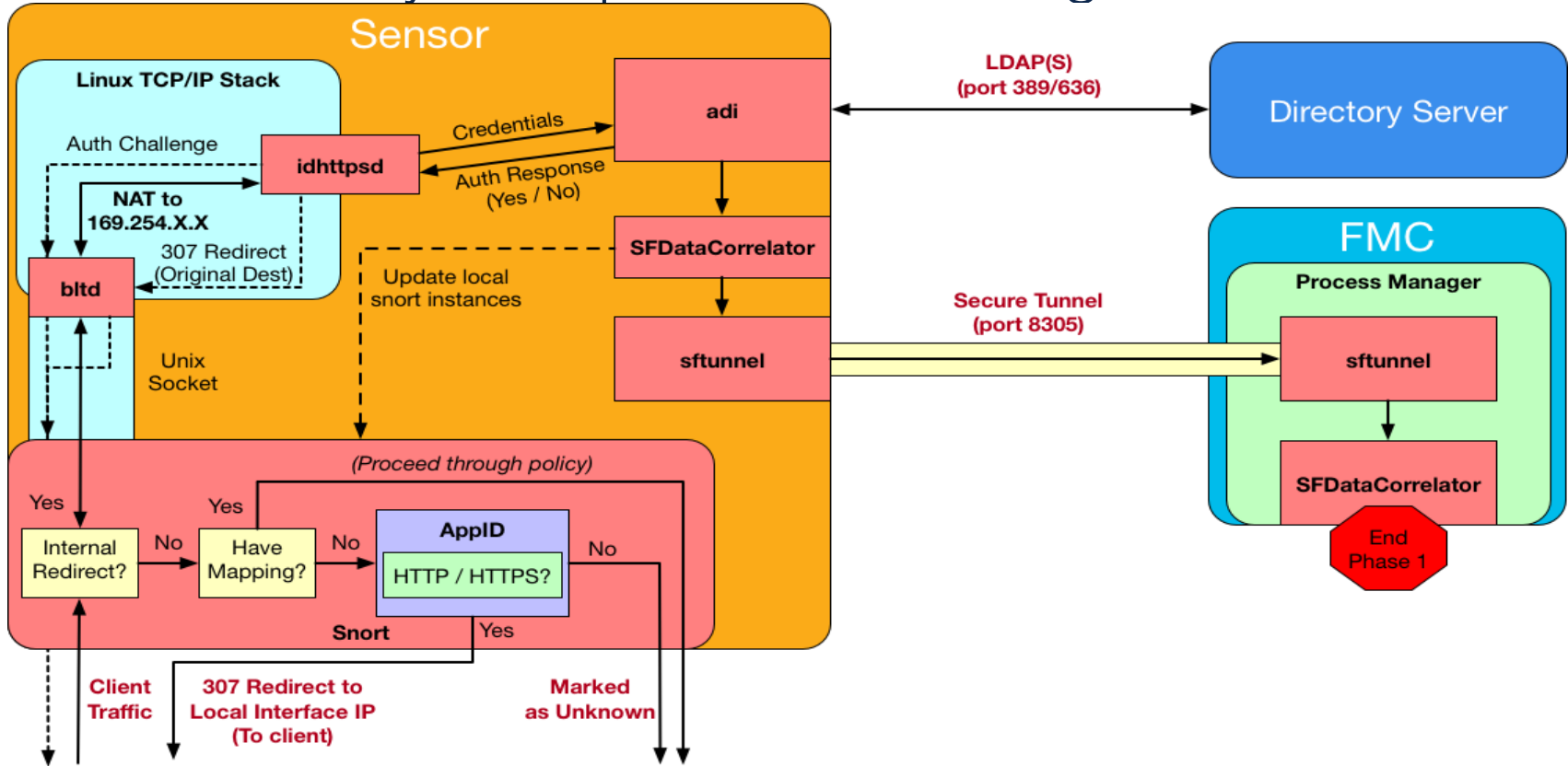
Snort Process Substeps

# Packet Processing: Captive Portal

- Will only act if traffic is identified as HTTP or HTTPS
- Evaluation point to see if a user mapping currently exists for this IP address
- Intercepts client traffic and forces them to authenticate if there is no active mapping



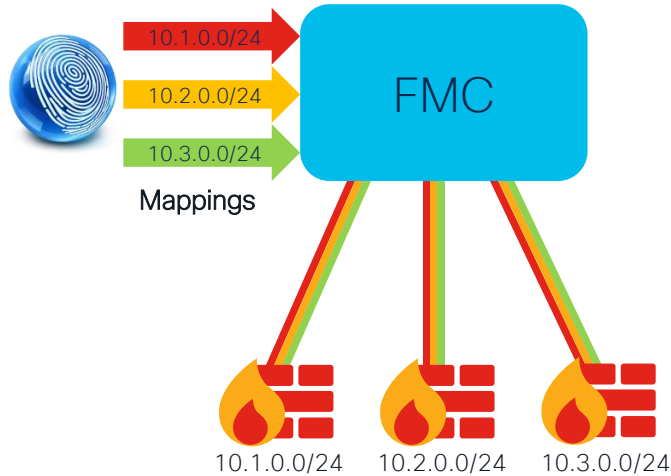
# User Identity - Captive Portal Diagram



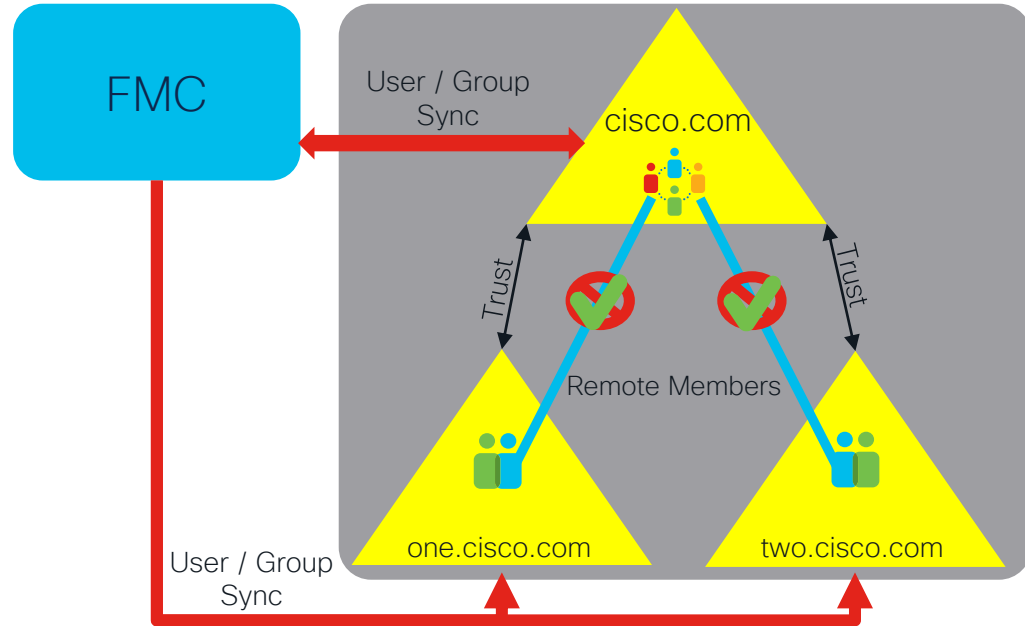
# Identity Feature Improvements

## Device-Level Identity Mapping Filter

\*CLI - 6.7 / GUI - 7.0



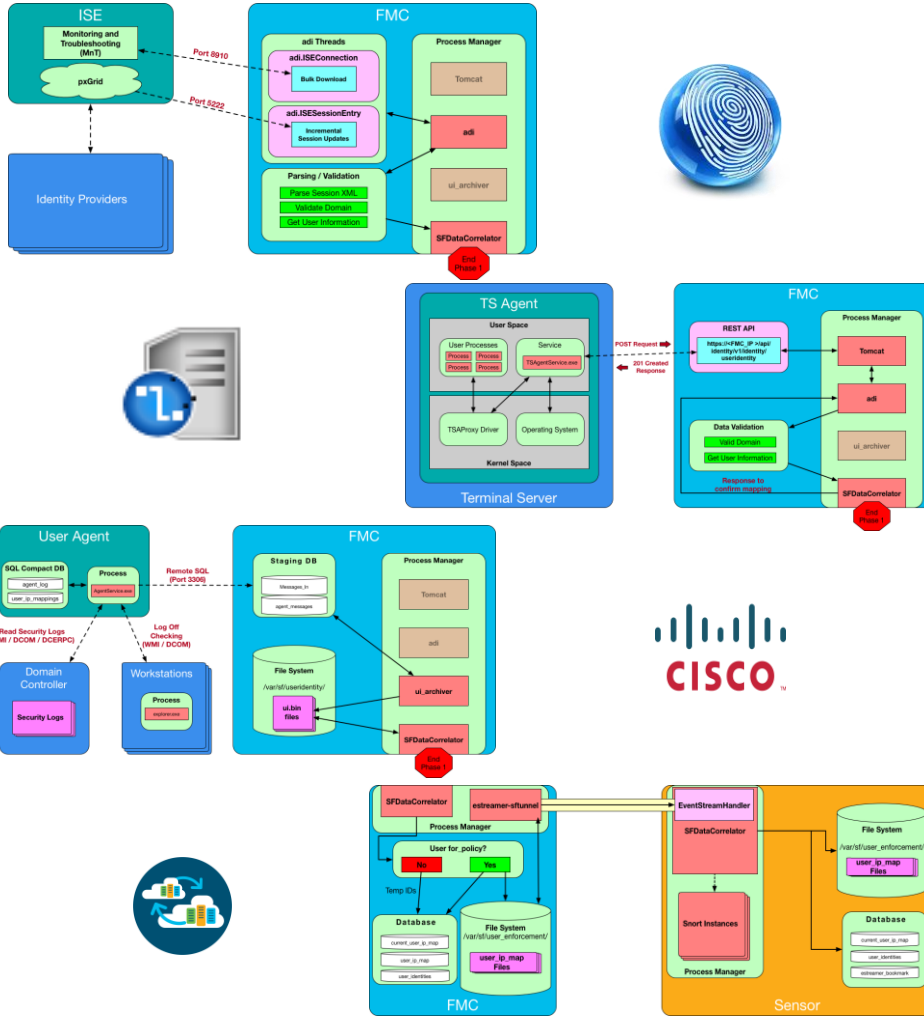
## Cross Domain Groups



# Want more on Identity?

BRKSEC-3227

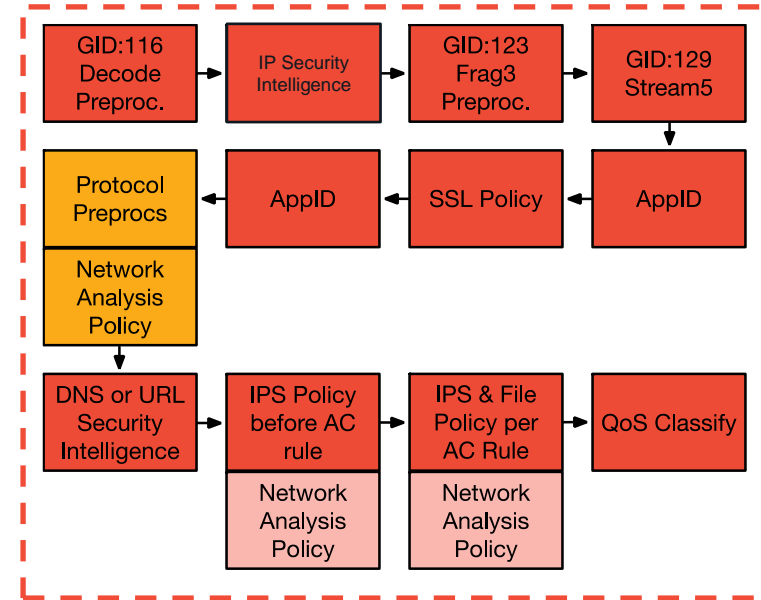
## Integrating & Troubleshooting Identity Features on the Firepower System



# Packet Processing: Protocol Preprocessors

Default Application Layer (L7) Preprocessors in a “Balanced Security and Connectivity” Network Analysis Policy (NAP):

Enabled	GID	Disabled	GID
DCE/RPC	133	SIP	140
DNS	131	IMAP	141
FTP & Telnet	125, 126	POP	142
HTTP	119		
Sun RPC	106		
GTP Command Channel	143		
SMTP	124		
SSH	128		
SSL	137		



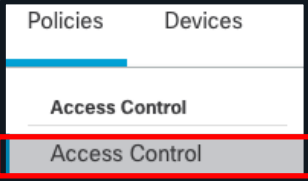
Snort Process Substeps

Not shown:


Transport and Network Layer, SCADA, Specific Threat preprocessors

# Packet Processing: Build a Network Analysis Policy


1



2

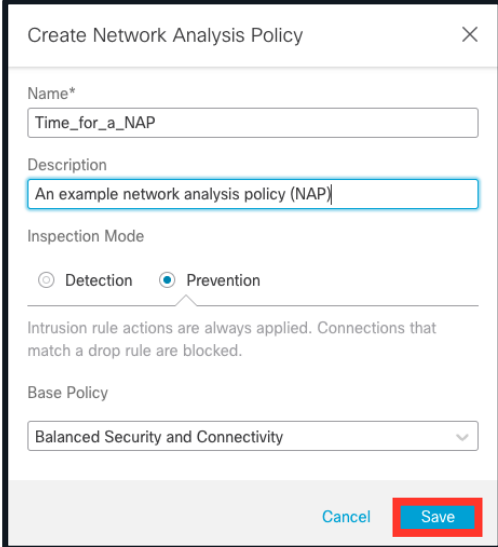


3



Create a Network Analysis Policy

4



Create Network Analysis Policy

Name\*  
Time\_for\_a\_NAP

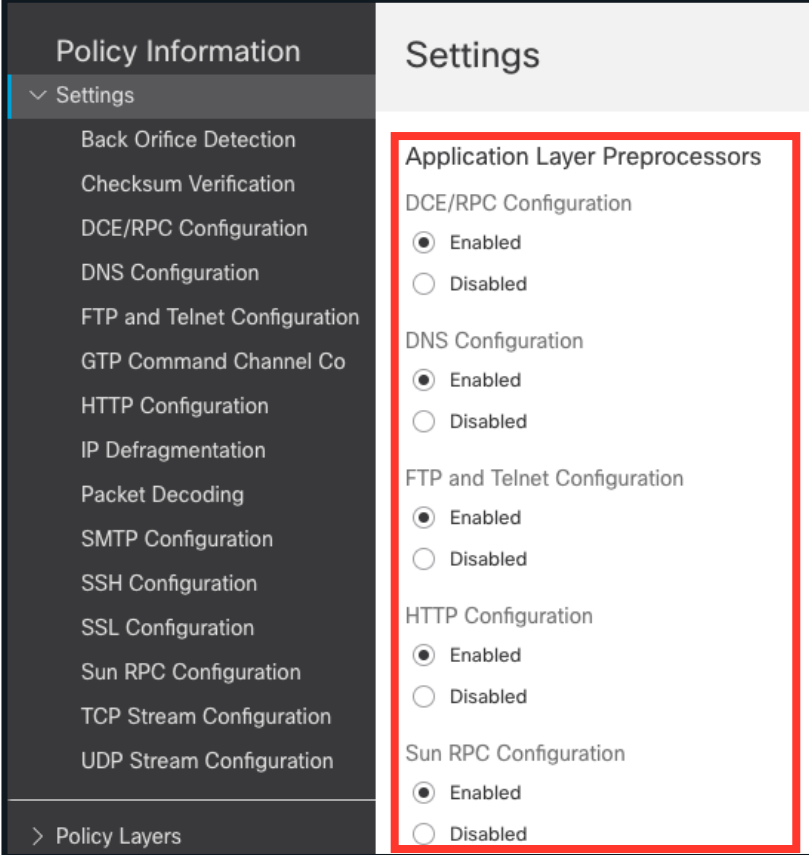
Description  
An example network analysis policy (NAP)

Inspection Mode  
 Detection  Prevention

Intrusion rule actions are always applied. Connections that match a drop rule are blocked.

Base Policy  
Balanced Security and Connectivity

Cancel Save



5

Policy Information

Settings

- Back Orifice Detection
- Checksum Verification
- DCE/RPC Configuration
- DNS Configuration
- FTP and Telnet Configuration
- GTP Command Channel Co
- HTTP Configuration
- IP Defragmentation
- Packet Decoding
- SMTP Configuration
- SSH Configuration
- SSL Configuration
- Sun RPC Configuration
- TCP Stream Configuration
- UDP Stream Configuration

> Policy Layers

Settings

Application Layer Preprocessors

DCE/RPC Configuration

- Enabled
- Disabled

DNS Configuration

- Enabled
- Disabled

FTP and Telnet Configuration

- Enabled
- Disabled

HTTP Configuration

- Enabled
- Disabled

Sun RPC Configuration

- Enabled
- Disabled



# Packet Processing: Apply a Network Analysis Policy

Rules Security Intelligence HTTP Responses Logging **Advanced** 1

TLS Server Identity Discovery ✎

Early application detection and URL categorization Disabled

Prefilter Policy Settings ✎

Prefilter Policy used before access control Default Prefilter Policy

**Network Analysis and Intrusion Policies** 2 ✎

Intrusion Policy used before Access Control rule is determined Balanced Security and Connectivity

Intrusion Policy Variable Set Default-Set

Default Network Analysis Policy Balanced Security and Connectivity

Network Analysis and Intrusion Policies ?

Intrusion Policy used before Access Control rule is determined  
Balanced Security and Connectivity ▾

Intrusion Policy Variable Set  
Default-Set ▾ ✎

Network Analysis Rules  
**No Custom Rules** [Network Analysis Policy List](#)

Default Network Analysis Policy  
Balanced Security and Connectivity ▾

[Revert to Defaults](#) [Cancel](#) [OK](#)

Network Analysis and Intrusion Policies ?

Intrusion Policy used before Access Control rule is determined  
Balanced Security and Connectivity ▾

Intrusion Policy Variable Set  
Default-Set ▾ ✎

Network Analysis Rules  
[1 Custom Rule](#) [Network Analysis Policy List](#)

**Add Rule** 4

#	Source Zones	Dest Zones	Source Networks	Dest Networks	VLAN Tags	Network Analysis Policy	
1	Any	Any	Any	Sleepy_Network	Any	Time_for_a_NAP	<span>✎</span> <span>🗑️</span>

Network Analysis Rules can help:

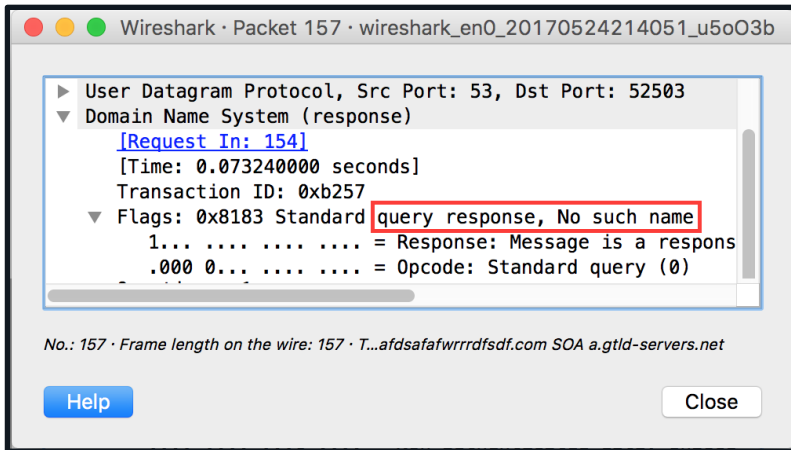
- Map a host / network segment to a custom NAP
- Exclude a host / network from default NAP

# Packet Processing: DNS Security Intelligence

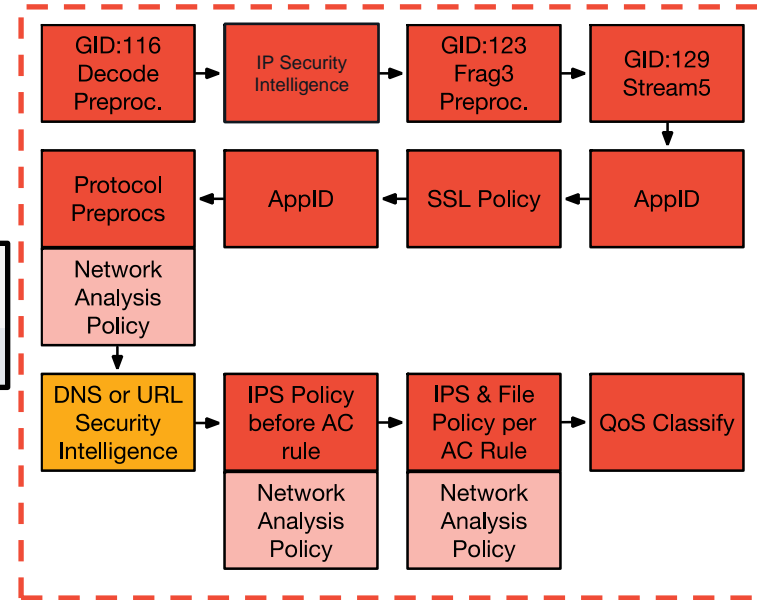
DNS SI performs a “man in the middle” of DNS queries

Option 1:

Alter DNS response to NXDOMAIN (domain not found)



NXDOMAIN Response



Snort Process Substeps

# Packet Processing: DNS Security Intelligence

## DNS Security Intelligence NXDomain - Firewall Engine Debug

```
> system support firewall-engine-debug
```

```
[lines removed]
```

```
10.1.1.2-54821 and 172.18.108.34-53 17 AS 1 I 1 no session DNS SI shared mem lookup  
returned 1 for example.com
```

```
[lines removed]
```


```
10.1.1.2-54821 and 172.18.108.34-53 17 AS 1 I 1 no session Got DNS list match. si list  
1048587  
10.1.1.2-54821 and 172.18.108.34-53 17 AS 1 I 1 no session Firing DNS action DNS NXDomain  
10.1.1.2-54821 and 172.18.108.34-53 17 AS 1 I 1 no session DNS SI: Matched rule order 3,  
Id 5, si list id 1048587, action 22, reason 2048, SI Categories 1048587,0
```

# Packet Processing: DNS Security Intelligence

DNS SI performs a “man in the middle” of DNS queries

Option 2:

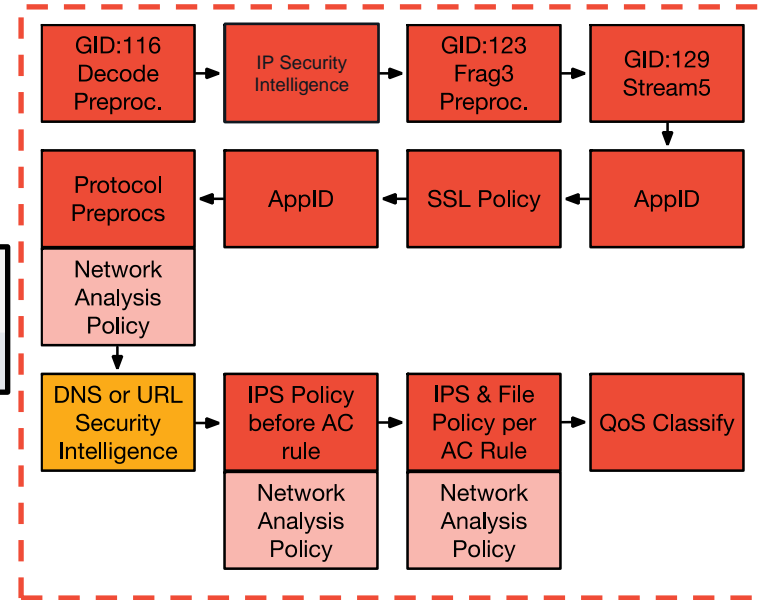
Alter DNS response to inject a Sinkhole server IP address



A configuration dialog box titled 'Sinkhole'. It contains the following fields and options:

- Name: CNC-Sinkhole
- IPv4 Policy: 10.99.99.99
- IPv6 Policy: 2001:DB8:1::dead:beef
- Log Connections to Sinkhole:
- Block and Log Connections to Sinkhole:
- Type: Command and Control

Buttons: Save, Cancel



Snort Process Substeps

# Packet Processing: DNS Security Intelligence

## DNS Security Intelligence Sinkhole - Firewall Engine Debug

```
> system support firewall-engine-debug
```

```
[lines removed]
```

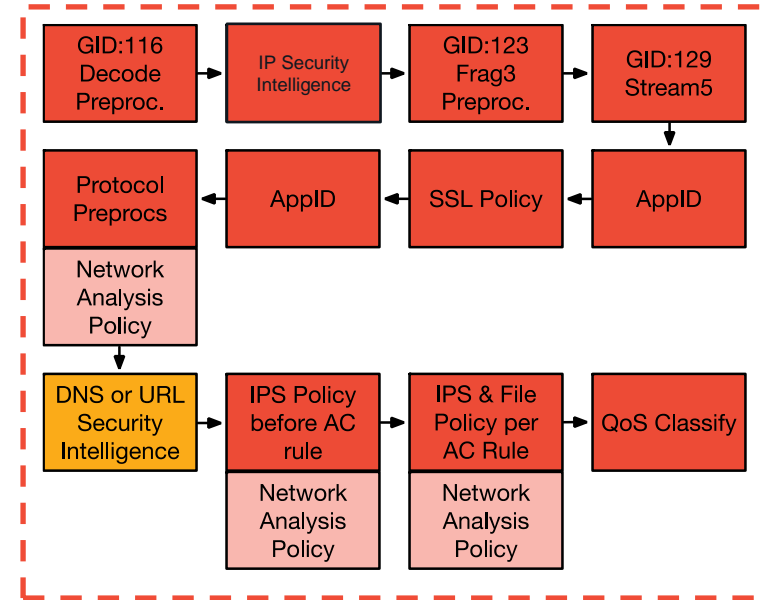
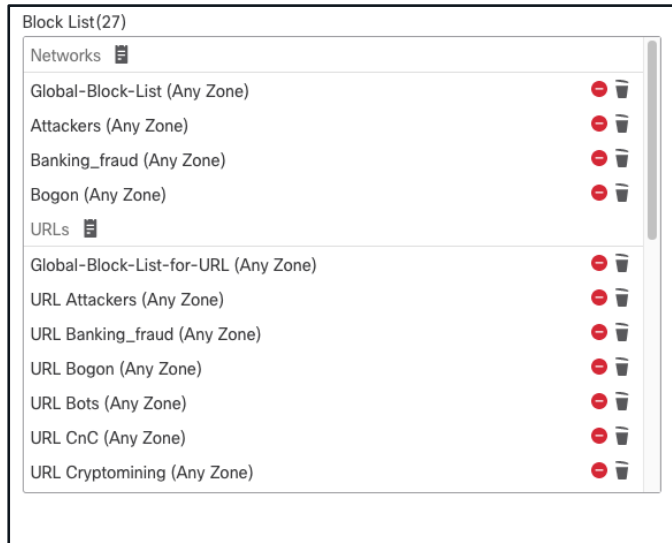
```
10.1.1.2-42818 and 172.18.108.34-53 17 AS 1 I 2 no session DNS SI shared mem lookup  
returned 1 for example.com
```

```
[lines removed]
```

```
10.1.1.2-42818 and 172.18.108.34-53 17 AS 1 I 2 no session Got DNS list match. si list  
1048587  
10.1.1.2-42818 and 172.18.108.34-53 17 AS 1 I 2 no session Firing DNS action DNS Sinkhole  
10.1.1.2-42818 and 172.18.108.34-53 17 AS 1 I 2 no session DNS SI: Matched rule order 3,  
Id 5, si list id 1048587, action 23, reason 2048, SI Categories 1048587,0
```

# Packet Processing: URL Security Intelligence

- URL SI is independent from Access Control URL rules
- Blocks lists of malicious domains
- Matches the HTTP GET or TLS Client Hello



Snort Process Substeps

# Packet Processing: URL Security Intelligence

URL Security Intelligence Block (Deny) - Firewall Engine Debug

```
> system support firewall-engine-debug
```

```
[lines removed]
```

```
10.1.1.2-35316 > 10.9.9.9-80 6 AS 1 I 21 URL SI:
```

```
ShmDBLookupURL("http://example.com/") returned 1
```

```
10.1.1.2-35316 > 10.9.9.9-80 6 AS 1 I 21 matched non-allow rule order 33, id 33
```

```
10.1.1.2-35316 > 10.9.9.9-80 6 AS 1 I 21 URL SI: Matched rule order 33, Id 33,  
si list id 1048584, action 4
```

```
10.1.1.2-35316 > 10.9.9.9-80 6 AS 1 I 21 deny action
```

# Packet Processing: URL Security Intelligence

## Dispute Reputations (6.5+)

Lookup data results for Domain: cisco.com

Search by IP, domain, or network owner for real-time threat data.

IP & Domain Reputation Overview | File Reputation Lookup | Email & Spam Data | Reputation Support

**LOCATION DATA**  
United States

**TOP CITIES**  
Allen, United States  
Richardson, United States  
Leidschendam, Netherlands  
Morrisville, United States  
San Jose, United States

**OWNER DETAILS**  
DOMAIN: cisco.com  
HOSTNAME: cisco.com

**MAIL SERVERS**  
rdn-mx-01.cisco.com  
alln-mx-01.cisco.com  
aer-mx-01.cisco.com

**REPUTATION DETAILS**  
WEB REPUTATION: Trusted  
LAST DAY: 5.4  
LAST MONTH: 5.4  
EMAIL VOLUME: 5.4  
VOLUME CHANGE: 7.84%

**BLOCK LISTS**  
TALOS SECURITY INTELLIGENCE BLOCK LIST  
ADDED TO BLOCK LIST: No

**CONTENT DETAILS**  
CONTENT CATEGORY: Computers and Internet  
Submit a Web Categorization Ticket

Submit a Web & Email Reputation Ticket

[https://talosintelligence.com/reputation\\_center](https://talosintelligence.com/reputation_center)



FMC UI (Analysis > Advanced > URL)

Overview | Analysis | Policies | Devices | Objects | AMP | Intelligence

Deploy | Search | Settings | Help | admin

URLs:(Limit 250)  
cisco.com  
Clear Search









URL	Category	Reputation	
cisco.com	Computers and Internet	Trusted	Dispute

Export CSV



# Packet Processing: URL Security Intelligence

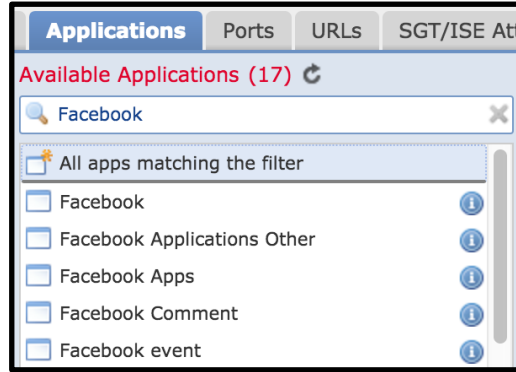
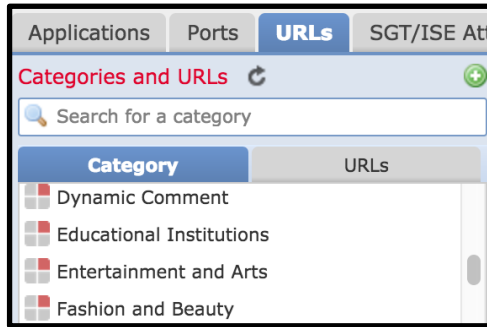
Analysis > Connections > Security Intelligence Events

<a href="#">First Packet</a> ×	<a href="#">Last Packet</a> ×	<a href="#">Action</a> ×	<a href="#">Reason</a> ×	<a href="#">Initiator IP</a> ×	<a href="#">Responder IP</a> ×	<a href="#">Security Intelligence</a> × <a href="#">Category</a>
<a href="#">2017-05-16 17:00:16</a>		<a href="#">Domain Not Found</a>	<a href="#">DNS Block</a>	 <a href="#">192.168.1.95</a>	 	<a href="#">DNS Response</a>
<a href="#">2017-05-16 16:57:50</a>	<a href="#">2017-05-16 16:57:50</a>	<a href="#">Block</a>	<a href="#">URL Block</a>	 <a href="#">192.168.1.95</a>	 <a href="#">10.83.48.40</a>	<a href="#">my_custom_url</a>
<a href="#">2017-05-16 16:50:05</a>		<a href="#">Block</a>	<a href="#">IP Block</a>	 <a href="#">192.168.1.95</a>	 	<a href="#">Malware</a>

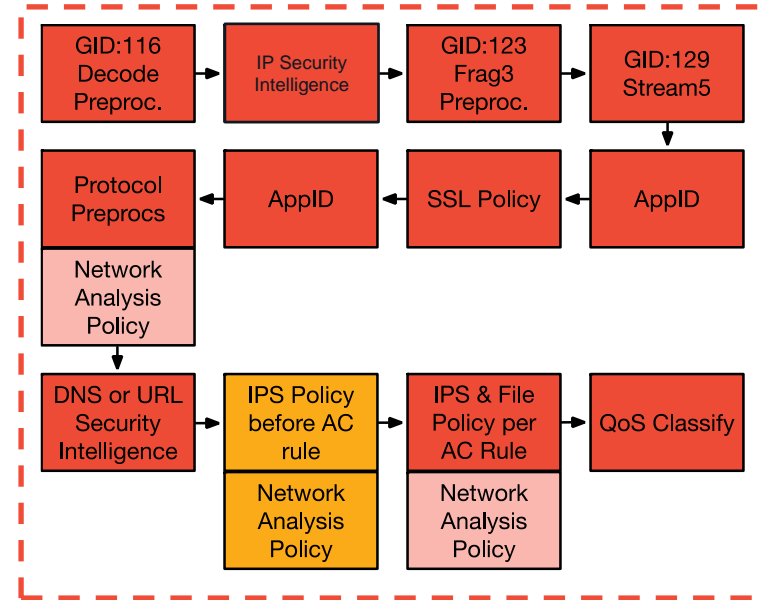
With logging enabled for all SI types you should be able to easily see what is being blocked by SI.

# Packet Processing: IPS Policy before Access Rules

- Access Control rules can match URLs or Applications



- To match a URL or App rule, Snort often needs the TLS Client Hello or HTTP GET
- Packets sent in a flow before matching an AC rule hit the “Intrusion Policy used before Access Control rule is determined”



Snort Process Substeps

# Packet Processing: Access Control Policy Rules

Access Control Policy rules are evaluated from top to bottom

**Allow** - Permit unless prohibited by an IPS or File Policy

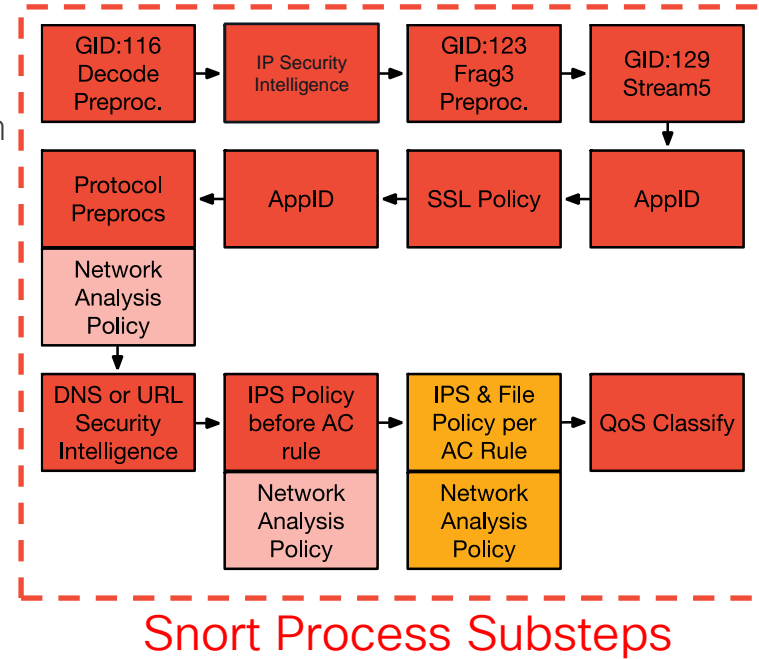
**Trust** - Pass the traffic without IPS or File inspection

**Block** - Silently drop the flow

**Block with Reset** - Send a TCP Reset or ICMP Unreachable

**Interactive Block with Reset** - Inject an HTTP 403 Forbidden

**Monitor** - Log the traffic and continue rule evaluation



# Packet Processing: Access Control Policy Rules

FMC

Analyze Hit Counts

Hit Count

Select a device:

jg-ftd-70.cisco.com

2-06-11 17:26:57

Prefilter (Default Prefilter Policy) → AC Policy (JG Policy)  
Total Rules 5

Filter Rules/Policy

Filter by: None

Last Deployed: 2022-06-08 13:39:48 (3 day(s) ago)

⚠ Policies have been modified and the hit count information displayed is not the latest.

#	Rule Name	Policy Name	Hit Count	Last Hit Time	
1	<a href="#">trust to gw</a>	JG Policy	0		🔍
2	<a href="#">block java</a>	JG Policy	0		🔍
3	<a href="#">block categories</a>	JG Policy	571	2022-06-11 16:55:48	🔍
4	<a href="#">omni inspect</a>	JG Policy	28454	2022-06-11 17:26:50	🔍
5	Default Action	JG Policy	0		

Displaying 1-5 of 5 rows | << Page 1 of 1 >> ↻

Generate CSV

Close

Security Policies

FDM

SSL Decryption → Identity → Security Intelligence → NAT → Access Control → Intrusion

1 rule

Search

Updated: less than 1 min ago

#	NAME	HIT COUNT LAST HIT	ACTION	SOURCE			DESTINATION			APPLICATIONS	URLS	USERS	ACTIONS
				ZONES	NETWORKS	PORTS	ZONES	NETWORKS	PORTS/PRO...				
> 1	Inside_Outside_Rule	3,476 2019-05-03 01:18:38	ANY	inside_zone	ANY	ANY	outside_zone	ANY	ANY	ANY	ANY	ANY	🔍

Default Action 0  
Not hit yet

Access Control **Block**

Hit Counters  
6.4

# Packet Processing: Access Control Rule Evaluation

#	Name	Sou... Zon...	Dest Zon...	Source Networks	Dest Networks	VLAN Tags	Users	Ap...	Sou... Ports	Dest Ports	URLs	Sou... Dyn... Attr...	Des... Dyn... Attr...	Action							
Mandatory - SSH Example (1-3)																					
1	Trust ssh for host	Any	Any	192.168.0.7	Any	Any	Any	Any	Any	SSH	Any	Any	Any	Trust						0	
2	inspect	Any	Any	10.0.0.0/8	Any	Any	Any	Any	Any	Any	Any	Any	Any	Allow						0	
3	trust server backup	Any	Any	192.168.62.3	10.123.175.22	Any	Any	Any	Any	Any	Any	Any	Any	Trust						0	

Original Client IP (HTTP)

SSH Connection from 192.168.62.3 to 10.123.175.22

1. SYN 192.168.62.3 → 10.123.175.22 Starts evaluation at 'inspect' rule
2. SYN,ACK 10.123.175.22 → 192.168.62.3
3. ACK 192.168.62.3 → 10.123.175.22
4. SSH 192.168.62.3 → 10.123.175.22



Pending AppID

Service identified as SSH  
No match 'inspect' rule (non-http)  
Match 'trust server backup' rule and Trust flow



# Packet Processing: Rule Evaluation

firewall-engine-  
debug



Example: SSH Connection from 192.168.62.3 to 10.123.175.22

SYN → SYN,ACK → ACK → First SSH Packet (client to server)

```
192.168.62.3-46594 > 10.123.175.22-22 6 AS 1 I 0 New session
192.168.62.3-46594 > 10.123.175.22-22 6 AS 1 I 0 Starting with minimum 4, 'inspect', and IPProto first
with zones 1 -> 2, geo 0 -> 0, vlan 0, inline sgt tag: untagged, ISE sgt id: 0, svc 0, payload 0, client
0, misc 0, user 9999997, icmpType 0, icmpCode 0
192.168.62.3-46594 > 10.123.175.22-22 6 AS 1 I 0 pending rule order 4, 'inspect', XFF wait for AppId
```

```
192.168.62.3-46594 > 10.123.175.22-22 6 AS 1 I 0 Starting with minimum 4, 'inspect', and IPProto first
with zones 1 -> 2, geo 0 -> 0, vlan 0, inline sgt tag: untagged, ISE sgt id: 0, svc 0, payload 0, client
0, misc 0, user 9999997, icmpType 0, icmpCode 0
192.168.62.3-46594 > 10.123.175.22-22 6 AS 1 I 0 pending rule order 4, 'inspect', XFF wait for AppId
```

```
192.168.62.3-46594 > 10.123.175.22-22 6 AS 1 I 0 Starting with minimum 4, 'inspect', and IPProto first
with zones 1 -> 2, geo 0 -> 0, vlan 0, inline sgt tag: untagged, ISE sgt id: 0, svc 0, payload 0, client
0, misc 0, user 9999997, icmpType 0, icmpCode 0
192.168.62.3-46594 > 10.123.175.22-22 6 AS 1 I 0 pending rule order 4, 'inspect', XFF wait for AppId
```

```
192.168.62.3-46594 > 10.123.175.22-22 6 AS 1 I 0 Starting with minimum 4, 'inspect', and IPProto first
with zones 1 -> 2, geo 0 -> 0, vlan 0, inline sgt tag: untagged, ISE sgt id: 0, svc 846, payload -1,
client 2000000846, misc 0, user 9999997, icmpType 0, icmpCode 0
192.168.62.3-46594 > 10.123.175.22-22 6 AS 1 I 0 no match rule order 4, 'inspect', XFF non-http
192.168.62.3-46594 > 10.123.175.22-22 6 AS 1 I 0 match rule order 5, 'trust server backup', action Trust
```

# Packet Processing: Rule Evaluation

firewall-engine-  
debug



SSH Connection from 192.168.62.3 to 10.123.175.22 (truncated)

```
192.168.62.3-46594 > 10.123.175.22-22 6 AS 1 I 0 New session
192.168.62.3-46594 > 10.123.175.22-22 6 AS 1 I 0 Starting with minimum 4, 'inspect', and IPProto first
with zones 1 -> 2, geo 0 -> 0, vlan 0, inline sgt tag: untagged, ISE sgt id: 0, svc 0, payload 0, client
0, misc 0, user 9999997, icmpType 0, icmpCode 0
192.168.62.3-46594 > 10.123.175.22-22 6 AS 1 I 0 pending rule order 4, 'inspect', XFF wait for AppId
```

[...omitted for brevity]

```
192.168.62.3-46594 > 10.123.175.22-22 6 AS 1 I 0 Starting with minimum 4, 'inspect', and IPProto first
with zones 1 -> 2, geo 0 -> 0, vlan 0, inline sgt tag: untagged, ISE sgt id: 0, svc 846, payload -1,
client 200000846, misc 0, user 9999997, icmpType 0, icmpCode 0
192.168.62.3-46594 > 10.123.175.22-22 6 AS 1 I 0 no match rule order 4, 'inspect', XFF non-http
192.168.62.3-46594 > 10.123.175.22-22 6 AS 1 I 0 match rule order 5, 'trust server backup', action Trust
```

[! How to map service/application ID to name]

```
> expert
$ grep "^846[^0-9]" /var/sf/appid/odp/appMapping.data
846 SSH 32 0 0 ssh
```

# Packet Processing: Rule Evaluation

firewall-engine-  
debug



SSH Connection from 192.168.62.3 to 10.123.175.22

(Blocked/Ended before matching an AC rule)

```
192.168.62.3-54650 > 10.123.175.22-22 6 AS 1 I 0 New session
192.168.62.3-54650 > 10.123.175.22-22 6 AS 1 I 0 Starting with minimum 4, 'inspect', and IPProto first with
zones 1 -> 2, geo 0 -> 0, vlan 0, inline sgt tag: untagged, ISE sgt id: 0, svc 0, payload 0, client 0, misc
0, user 9999997, icmpType 0, icmpCode 0
192.168.62.3-54650 > 10.123.175.22-22 6 AS 1 I 0 pending rule order 4, 'inspect', XFF wait for AppId
192.168.62.3-54650 > 10.123.175.22-22 6 AS 1 I 0 Deleting session
```

```
[!Session was deleted because we hit a drop IPS rule and blocklisted the flow.
This happened before AC rule was matched (Intrusion policy before AC rule match dropped).
Firewall engine will re-evaluate from top of AC policy to find a rule for logging decision]
```

```
192.168.62.3-54650 > 10.123.175.22-22 6 AS 1 I 0 Starting with minimum 0, id 0 and IPProto first with zones
1 -> 2, geo 0 -> 0, vlan 0, inline sgt tag: 0, ISE sgt id: 0, svc -1, payload -1, client -1, misc -1, user
9999997, icmpType 102, icmpCode 22
192.168.62.3-54650 > 10.123.175.22-22 6 AS 1 I 0 no match rule order 3, 'Trust ssh for host', src network
and GEO
192.168.62.3-54650 > 10.123.175.22-22 6 AS 1 I 0 no match rule order 4, 'inspect', XFF non-http
192.168.62.3-54650 > 10.123.175.22-22 6 AS 1 I 0 match rule order 5, 'trust server backup', action Trust
```

Action ×	Reason ×	Initiator IP ×	Responder IP ×	Source Port / ICMP Type ×	Destination Port / ICMP Code ×	Application Protocol ×	Client ×	Intrusion Events ×	Access Control Policy ×	Access Control Rule ×
Block	Intrusion Block	192.168.62.3	10.123.175.22	55654 / tcp	22 (ssh) / tcp				JG AC (all)	trust server backup

AC Rule has “Trust” action but connection event action shows “Block”



# Packet Processing: Access Control with IPS

The screenshot displays the Cisco Talos IPS configuration interface. On the left is a dark sidebar with a navigation menu. The main area shows the configuration for a specific rule.

**Policy Information**

- Rules
- Firepower Recommendations
- Advanced Settings

**Policy Layers**

- My Changes (highlighted with a red box)
- Rules
- Balanced Security and Conn
- Rules
- Global Rule Thresholding

**Rules - Base Policy** < Back

Rule Configuration

Rule Content

Filter:

0 selected rules of 1

<input type="checkbox"/>	GID	SID	Message	→	⌕	⏸	ⓘ	🗨
<input checked="" type="checkbox"/>	1	31977	OS-OTHER Bash CGI environment variable injection attempt	×				

[Hide details](#) Above Below ⏪ < 1 of 1 > ⏩

(1:31977) OS-OTHER Bash CGI environment variable injection attempt

Rule State  Drop and Generate Events

Firepower Recommendation

Rule Overhead

- > Thresholds (0)
- >Suppressions (0)
- > Dynamic State (0)
- > Alerts (0)
- > Comments (0)
- > Documentation

Category

- Classifications
- Microsoft Vulnerabilities
- Microsoft Worms

rule

```
alert tcp $EXTERNAL_NET any -> $HOME_NET $HTTP_PORTS (msg:"OS-OTHER Bash CGI environment variable injection attempt"; flow:to_server,established; content:"{}"; fast_pattern:only; http_uri; metadata:policy balanced-ips drop, policy max-detect-ips drop, policy security-ips drop, ruleset community, service http; reference:cve.2014-6271; reference:cve.2014-6277; reference:cve.2014-6278; reference:cve.2014-7169; classtype:attempted-admin; sid:31977; rev:5; gid:1;)
```

Intrusion Policies are built on layers

Prebuilt base layers from Cisco **TALOS**

- Connectivity over Security (~500 rules)
- Balanced Security & Connectivity (~9,400 rules)
- Security over Connectivity (~20,300 rules)

# Packet Processing: Access Control with File

Application Protocol: Any

Action: Block Malware

Store Files:  Malware,  Unknown,  Clean,  Custom

Direction of Transfer: Any

Spero Analysis for MSEXE

Dynamic Analysis

Capacity Handling

Local Malware Analysis

Reset Connection

File Type Categories: Office Documents (18), Archive (19), Multimedia (4)

File Types: Search name and description, 7Z (7-Zip compressed file), ACCDB (Microsoft Access ...)

Selected File Categories and Types: Category: PDF files, Category: Executables, Category: Office Documents

Add

- Like Intrusion Policies, a **File Policy** is tied to an Access Control Rule
- Checks files by looking at the **SHA256** hash to compare against known malware hashes
- Can submit unknown files to the **AMP** cloud or **Secure Malware Analytics (SMA)** appliance

```
> system support firewall-engine-debug
10.1.1.2-16969 > 10.9.9.9-80 6 AS 0 I 1 File malware event for
275a021bbfb6489e54d471899f7db9d1663fc695ec2fe2a2c4538aabf651fd0f named eicar.com with
disposition Malware and action Block Malware
```

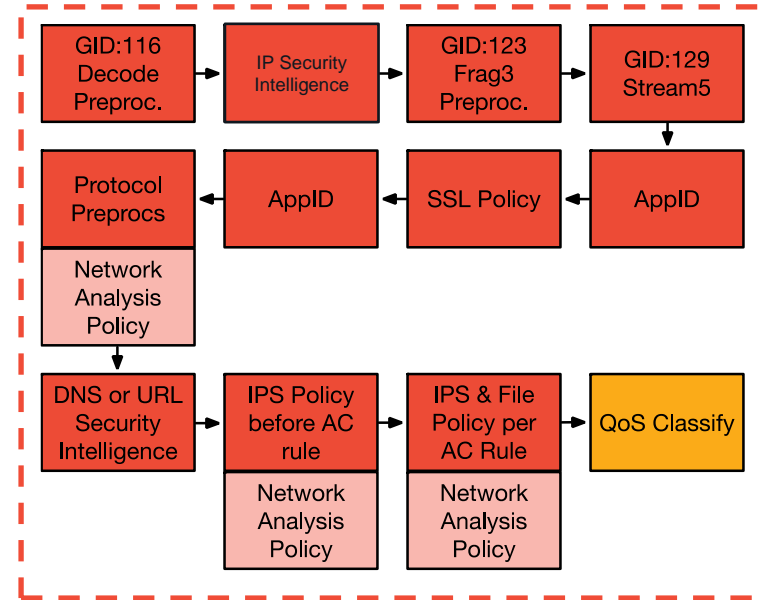
# Packet Processing: QoS Classification in Snort

Eligible traffic for rate-limiting:

- Allowed or Trusted

Ineligible traffic for rate-limiting:

- Blocked or Prefilter Fastpath (Snort exempt)
- Snort classifies traffic by matching it to a QoS rule
- Snort tells Lina the Flow-rule QoS id for each flow
- On the Lina interface, the Rule ID matches a traffic class



Snort Process Substeps

# Packet Processing: QoS Classification in Snort

Rules Policy Assignments (1)

[Filter by Device](#) + Add Rule

#	Name	Source Interface Objects	Dest Interface Objects	Source Netw...	Dest Netw...	Users	Applic...	Source Ports	Dest Ports	UR...	Source SGT	Rate Limit per Interface			Applied On	🗨	🗑
												Download	Upload				
1	Police HTTP (80)	inside	Any	10.0.0.0/8	Any	Any	Any	TCP (6):80	Any	Any	Any	1 Mbits/sec	Unlimited	Source interfa	0		

```
> expert
$ cat /ngfw/var/sf/detection_engines/[UUID]/qos.rules
[lines removed]
268435467 ratelimit 2 10.0.0.0 8 any any any 80 any 6 ← QoS Rule ID
> system support firewall-engine-debug
[lines removed]
10.1.1.2-59831 > 10.9.9.9-80 6 AS 1 I 19 match rule order 1, id 268435467 action Rate Limit
10.1.1.2-59831 > 10.9.9.9-80 6 AS 1 I 19 QoS policy match status ((null)), match action
(Rate Limit), QoS rule id (268435467)
```

# Packet Processing: QoS Interface Policing in LINA

```
> system support diagnostic-cli
```

```
firepower# show run service-policy
```

```
service-policy global_policy global
```

```
service-policy policy_map_inside interface inside
```

```
firepower# show service-policy interface inside
```

```
Interface inside:
```

```
Service-policy: policy_map_inside
```

```
Flow-rule QoS id: 268435467
```

← QoS Rule ID

```
Output police Interface inside:
```

```
  cir 1000000 bps, bc 31250 bytes
```

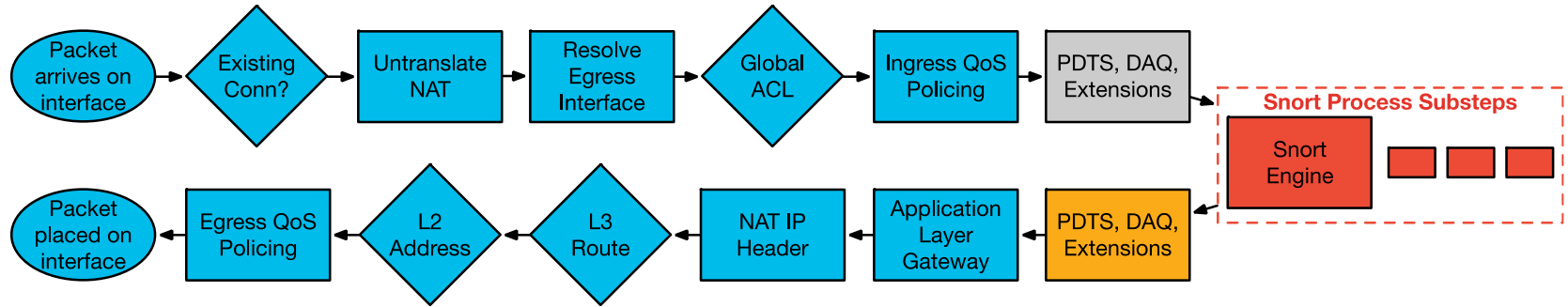
```
firepower# show conn detail
```

```
TCP outside: 10.9.9.9/80 inside: 10.1.1.2/59831,
```

```
  flags UxIO N, qos-rule-id 268435467, idle 0s, uptime 4m5s, timeout 1h0m, bytes
```

```
15542738, xlate id 0x7f05a30260c0
```

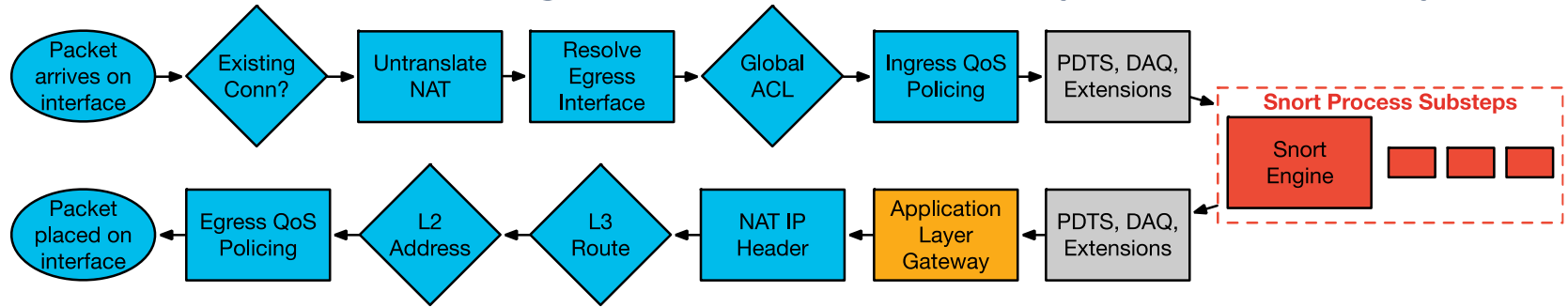
# Packet Processing: Packet Data Transport System



The Packet Data Transport System sends packets back to Lina after Snort processing.

Note: It is extremely rare for any packets to be dropped at this stage.

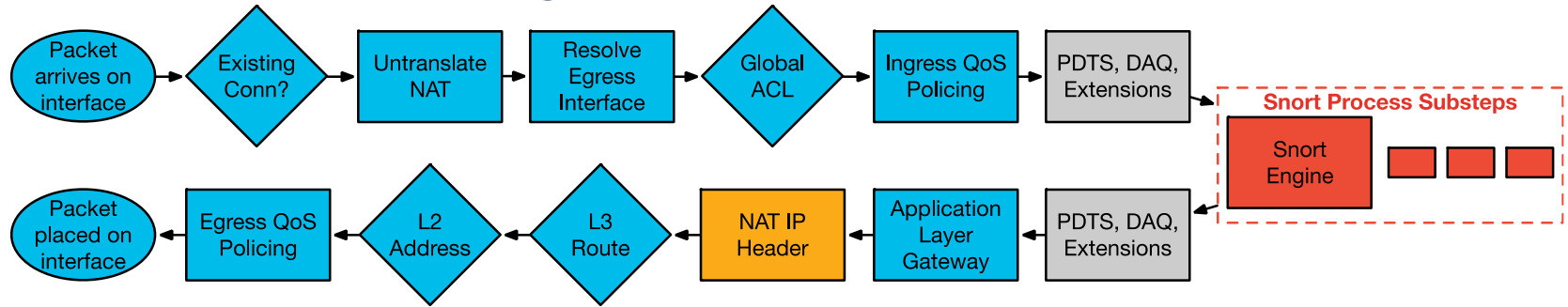
# Packet Processing: Application Layer Gateway



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

```
ASA-4-406002: FTP port command different address: 10.2.252.21(192.168.1.21) to  
209.165.202.130 on interface inside  
ASA-4-405104: H225 message received from outside_address/outside_port to  
inside_address/inside_port before SETUP
```

# Packet Processing: NAT IP Header



- Translate the source and destination IP addresses in the IP header
- Translate the port if performing PAT
- Update header checksums
- NAT rules are presented in a single table divided into categories
- NAT rules in the table are applied on a top-down, first-match basis.



# Auto NAT (Object NAT)

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

## Static Host NAT

```
object network obj-WebServer
  host 10.1.2.100
  nat (inside,outside) static 198.51.100.10
```

## Dynamic PAT (interface overload)

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

					Original Packet			Translated	
#	Direction	Type	Source Interface Objects	Destination Interface Objects	Original Sources	Original Destinations	Original Services	Translated Sources	Translated Destinations
> NAT Rules Before									
v Auto NAT Rules									
#..	↔	Static	Inside	Outside	10.1.2.100			198.51.100.1	
#..	↔	Dynamic	Inside	Outside	InternalUsers			Interface	

# Manual NAT (Twice NAT)

- Manual NAT can specify the source and the destination translations

## 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
```

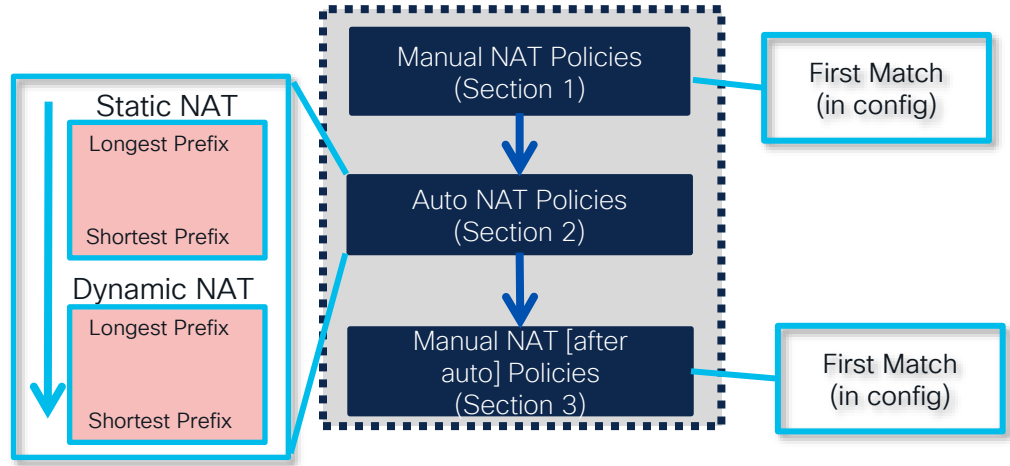
# NAT Order of Operation

- FTD configuration is built into the NAT table

```
> show nat
Manual NAT Policies (Section 1)
1 (Inside) to (Outside) source static SERVER OBJ-
192.168.20.10
   translate_hits = 0, untranslate_hits = 0

Auto NAT Policies (Section 2)
1 (nlp_int_tap) to (Inside) source static
nlp_server_0_ssh_intf6 interface service tcp ssh ssh
   translate_hits = 0, untranslate_hits = 0

Manual NAT Policies (Section 3)
1 (Inside) to (Outside) source dynamic Inside-Network
interface
   translate_hits = 0, untranslate_hits = 0
```



# NAT Order of Operation

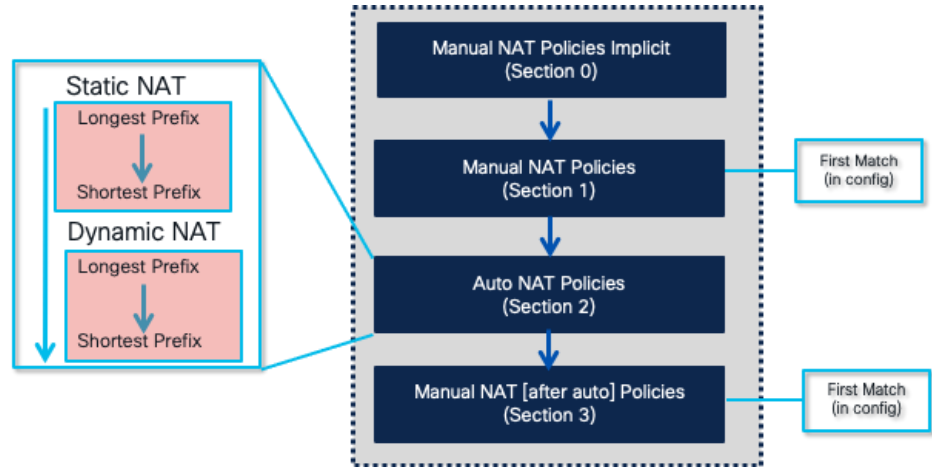


- In Firepower version 7.0, a new section, Section 0, is added to the NAT table for all implicit NAT rules for NLP applications (sftunnel, SSH, SNMP, HTTP)

```
> show nat
Manual NAT Policies Implicit (Section 0)
1 (nlp_int_tap) to (Inside) source static
nlp_server_ssh_0.0.0.0_intf2 interface destination
static 0.0.0.0_2 0.0.0.0_2 service tcp ssh ssh
translate_hits = 0, untranslate_hits = 0

Manual NAT Policies (Section 1)
1 (Inside) to (Outside) source static SERVER OBJ-
192.168.20.10
translate_hits = 0, untranslate_hits = 0

Manual NAT Policies (Section 3)
1 (Inside) to (Outside) source dynamic Inside-Network
interface
translate_hits = 0, untranslate_hits = 0
```

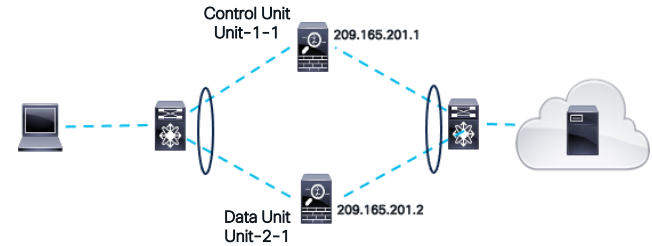


Want to learn more about  
NAT/PAT in Secure  
Firewall?

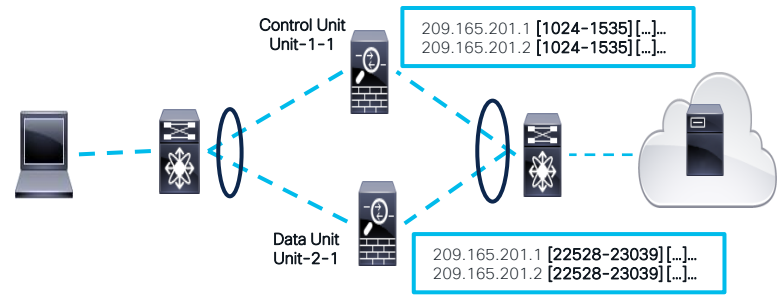
## BRKSEC-2102

Firepower Cluster NAT and  
PAT Operation and  
Troubleshooting

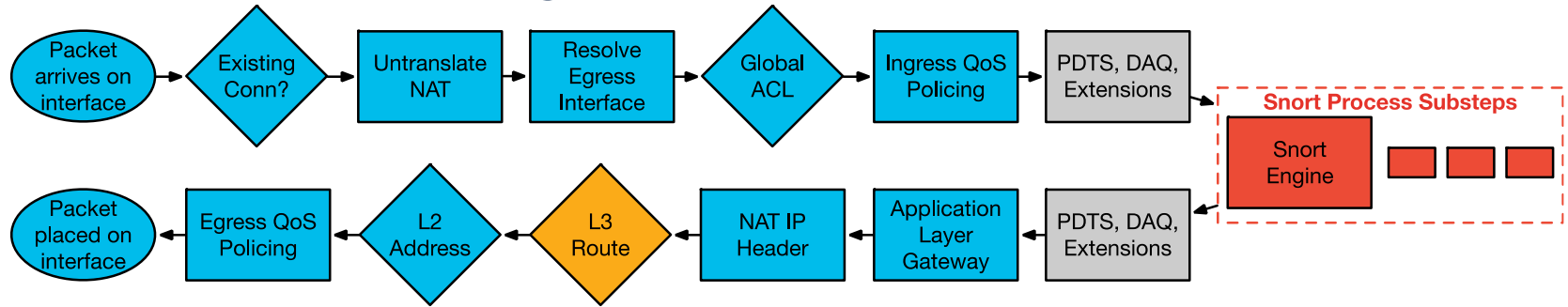
Firepower 6.6 and Earlier:



Firepower 6.7.+



# Packet Processing: L3 Route Lookup



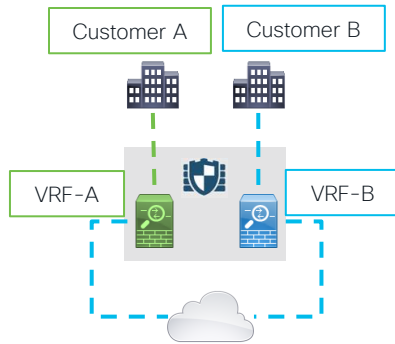
- After the IP header translation an interface route lookup is performed
- Only routes pointing out the egress interface are eligible
- Remember: NAT rule can forward the packet to the egress interface, even though the routing table may point to a different interface
  - If the destination is not routable 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
```

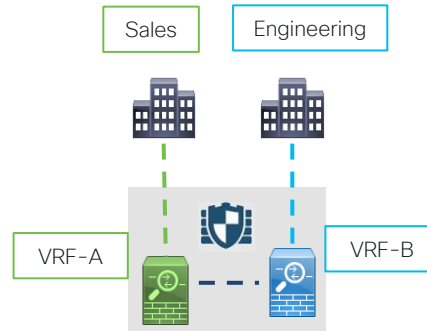


# Virtual Routing and Forwarding

- Routing segregation on FTD
- VRF-Lite
- Overlapping IP Address on FTD interfaces across Virtual Routers
- Use cases:



Service Provider



Enterprise  
(Route Leaking)

# Virtual Routing and Forwarding



Cluster Device **Routing** Interfaces Inline Sets DHCP

Manage Virtual Routers

VRF-Sales

Virtual Router Properties

These are the basic details of this virtual router.

VRF Name: VRF-Sales

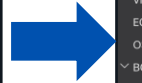
Description:

Select Interface:

Available Interfaces C

Selected Interfaces

Add



Manage Virtual Routers

VRF-Sales

Virtual Router Properties

ECMP

OSPF

BGP

IPV4

IPV6

Static Route

Network	Interface	Leaked from Virtual Router	Gateway	Tunneled
▼ IPv4 Routes				
any-ipv4	Outside	Global		false
▼ IPv6 Routes				

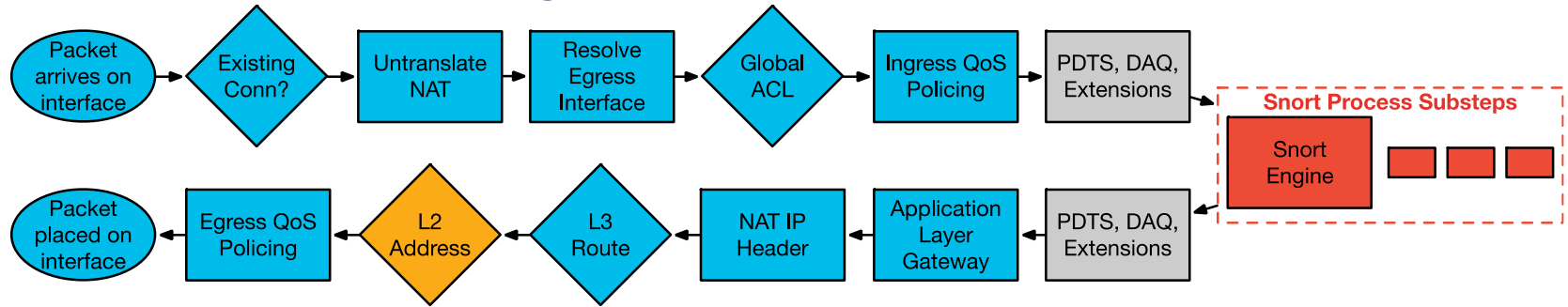
```
> show route all
Gateway of last resort is not set
C    209.165.201.0 255.255.255.0 is directly connected, Outside
L    209.165.201.3 255.255.255.255 is directly connected, Outside
Routing Table: VRF-Sales
Gateway of last resort is 0.0.0.0 to network 0.0.0.0

SI  0.0.0.0 0.0.0.0 [1/0] is directly connected, Outside
C    192.168.10.0 255.255.255.0 is directly connected, Inside
L    192.168.10.1 255.255.255.255 is directly connected, Inside
```

```
> packet-tracer input inside icmp 192.168.10.1 8 0 209.165.201.10
Phase: 3
Type: IMPORTED-ROUTE
Subtype: vrf imported route
Result: ALLOW
Elapsed time: 5137 ns
Config:
Additional Information:
in 0.0.0.0 0.0.0.0 via 0.0.0.0, Outside (Imported Route)-12
Phase: 4
Type: INPUT-ROUTE-LOOKUP
Subtype: Resolve Egress Interface
Result: ALLOW
Elapsed time: 467 ns
Config:
Additional Information:
Found next-hop 0.0.0.0 using egress ifc Outside(vrfid:0)
```



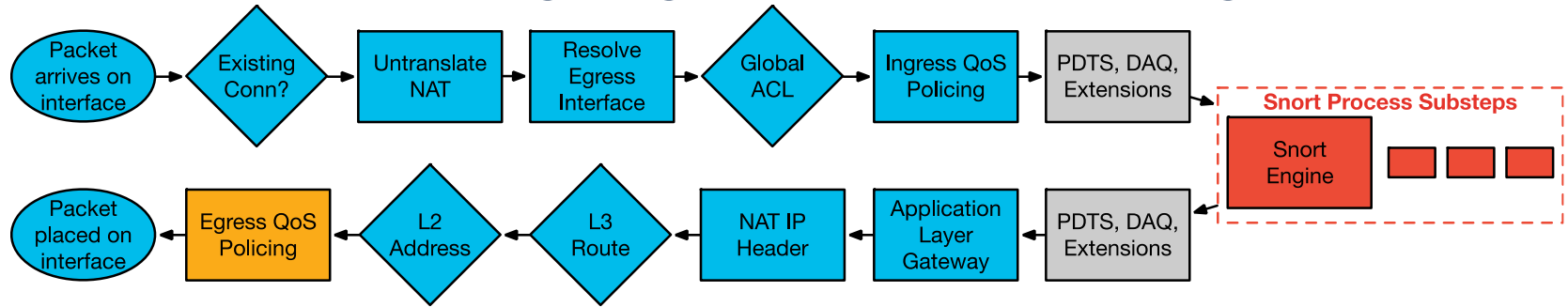
# Packet Processing: L2 Address Lookup



- 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: Egress QoS Policing



```
> system support diagnostic-cli
```

```
firepower# show service-policy interface outside
```

```
Interface outside:
```

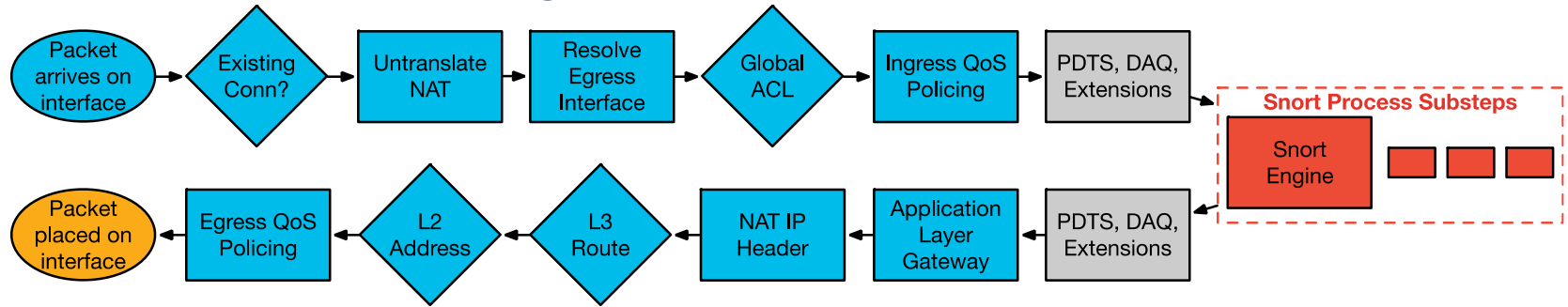
```
Service-policy: policy_map_outside
```

```
Flow-rule QoS id: 268435467
```

```
Output police Interface outside:
```

```
cir 1000000 bps, bc 31250 bytes
```

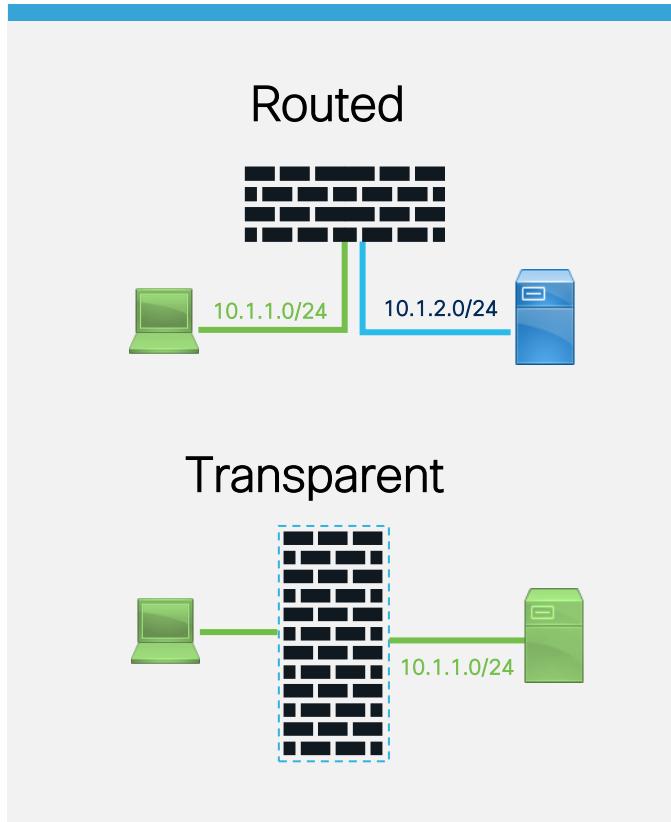
# Packet Processing: Transmit Packet



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

```
> show interface outside
Interface GigabitEthernet0/1 "outside", is up, line protocol is up
...
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)
```

# Packet Processing: FTD Deployment Modes

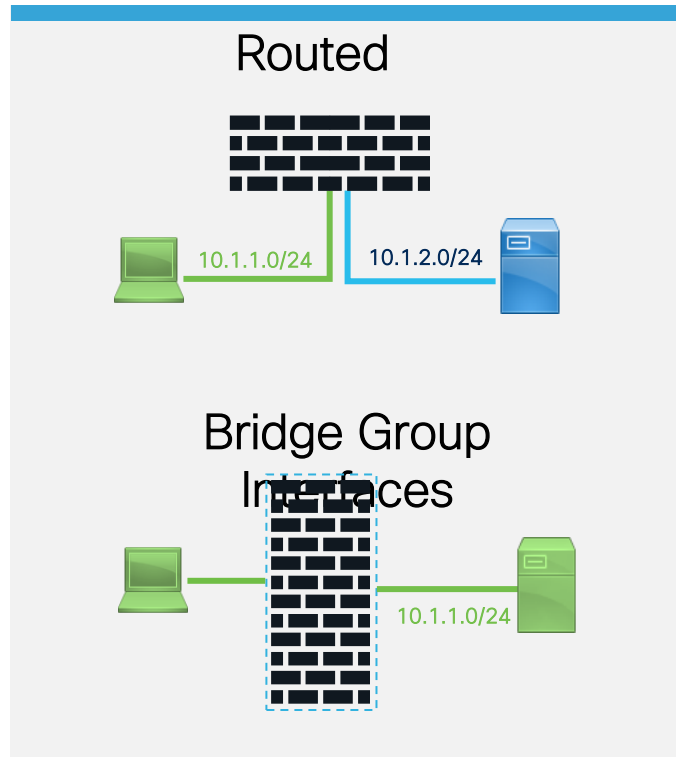


- FTD is considered to be Layer 3 device
- FTD can route traffic between different subnets

- FTD is considered to be a Layer 2 device
- Interfaces are defined within a Bridge Group
  - A Bridge group represents a unique Layer 2 network
  - Re-writes VLAN tags in trunk mode

# Packet Processing: FTD Interface Modes

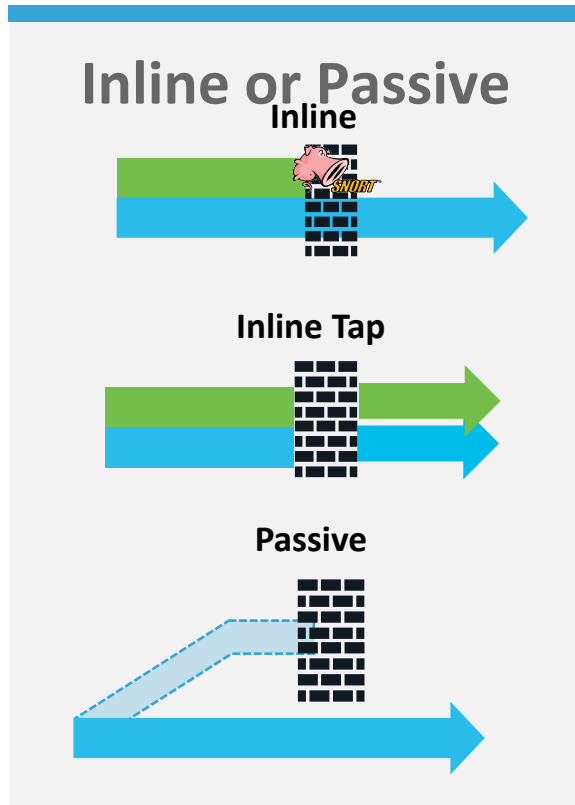
## Regular Firewall Mode



- Routed firewall mode only
  - Layer 3 interfaces.
    - Each interface is on a different subnet
- 
- Routed and Transparent firewall mode
  - Interfaces are defined within a Bridge Group

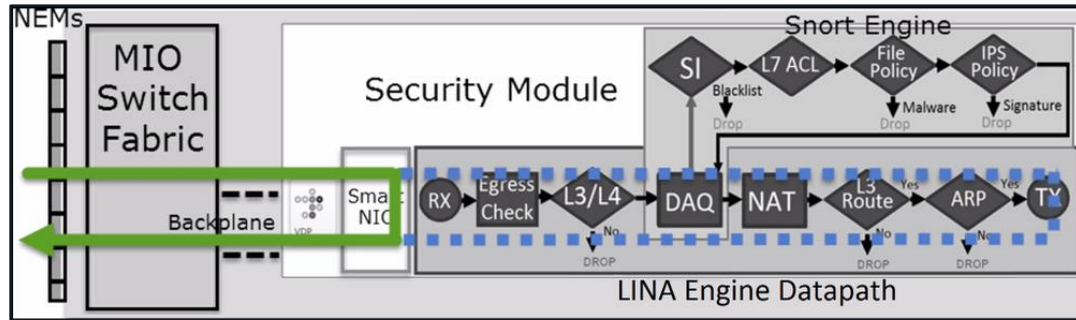
# Packet Processing: FTD Interface Modes

## IPS-Only Mode



- Functions as an L1 “bump in the wire”, no L2/L3 packet re-writing
- Snort processing only (Lina sees the packet but only redirects to Snort)
- A copy of each packet is sent to snort for inspection.
- Passive interfaces monitor traffic flowing across a network using a switch SPAN or mirror port.

# Packet Processing: Flow Offload



- Bypasses Lina and Snort completely
  - L2/L3 re-writing is handled by special network adapter in the security engine blade
  - View offloaded flows via the 'show flow-offload flow detail' command in Lina CLI
1. Static Flow Offload:
    - Connections that are fastpathed by the prefilter policy.
  2. Dynamic Flow Offload:
    - Inspected flows that the inspection engine decides no longer need inspection.

# Access control Rule Explosion



#	Name	Source Zones	Dest Zones	Source Networks	Dest Networks	Action
Mandatory - ACP2 (1-1)						
1	Allow-Egress	InternalZones	ExternalZones	Source-hosts	Destination-hosts	Allow

InternalZones

- FTD-Cluster
- DMZ
- Inside

ExternalZones

- FTD-Cluster
- ISP-1

Name	Value
Source-hosts	10.10.10.2 10.10.10.1

Name	Value
Destination-hosts	20.20.20.2 20.20.20.1

$\text{InternalZones}(2) \times \text{ExternalZones}(1) \times \text{SourceHosts}(2) \times \text{DestinationHosts}(2) = 8 \text{ ACES}$

```
> show access-list
access-list CSM_FW_ACL_line 10 advanced permit ip ifc DMZ object-group Source-hosts ifc ISP-1 object-group Destination-hosts
access-list CSM_FW_ACL_line 10 advanced permit ip ifc DMZ host 10.10.10.2 ifc ISP-1 host 20.20.20.1 rule-id 268434437
access-list CSM_FW_ACL_line 10 advanced permit ip ifc DMZ host 10.10.10.2 ifc ISP-1 host 20.20.20.2 rule-id 268434437
access-list CSM_FW_ACL_line 10 advanced permit ip ifc DMZ host 10.10.10.1 ifc ISP-1 host 20.20.20.1 rule-id 268434437
access-list CSM_FW_ACL_line 10 advanced permit ip ifc DMZ host 10.10.10.1 ifc ISP-1 host 20.20.20.2 rule-id 268434437
access-list CSM_FW_ACL_line 11 advanced permit ip ifc Inside object-group Source-hosts ifc ISP-1 object-group Destination-hosts
access-list CSM_FW_ACL_line 11 advanced permit ip ifc Inside host 10.10.10.2 ifc ISP-1 host 20.20.20.1 rule-id 268434437
access-list CSM_FW_ACL_line 11 advanced permit ip ifc Inside host 10.10.10.2 ifc ISP-1 host 20.20.20.2 rule-id 268434437
access-list CSM_FW_ACL_line 11 advanced permit ip ifc Inside host 10.10.10.1 ifc ISP-1 host 20.20.20.1 rule-id 268434437
access-list CSM_FW_ACL_line 11 advanced permit ip ifc Inside host 10.10.10.1 ifc ISP-1 host 20.20.20.2 rule-id 268434437
```

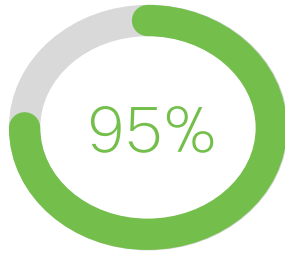




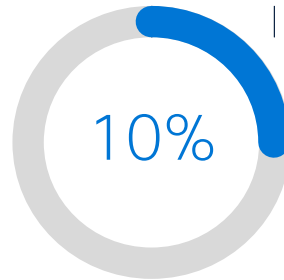
# Access Control Rule Optimization

## Object Group Search (OGS)

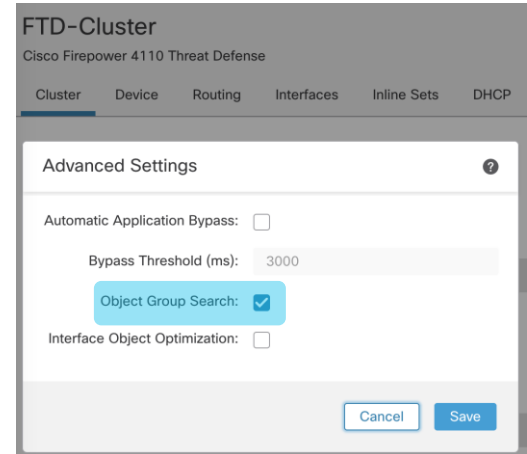
- Access Control List optimization feature on FTD 6.6+
- It will install just one rule, instead of expanding the Access Control Elements
- It might increase CPU usage during during packet processing



Memory usage  
reduction



Deploy time  
reduction





# Access Control Rule Optimization

## Object Group Search (OGS)

- Rule expansion with OGS disabled.

```
> show access-list
access-list CSM_FW_ACL_ line 10 advanced permit ip ifc DMZ object-group Source-hosts ifc ISP-1 object-group Destination-hosts rule-id
268434437
access-list CSM_FW_ACL_ line 10 advanced permit ip ifc DMZ host 10.10.10.2 ifc ISP-1 host 20.20.20.1 rule-id 268434437
access-list CSM_FW_ACL_ line 10 advanced permit ip ifc DMZ host 10.10.10.2 ifc ISP-1 host 20.20.20.2 rule-id 268434437
access-list CSM_FW_ACL_ line 10 advanced permit ip ifc DMZ host 10.10.10.1 ifc ISP-1 host 20.20.20.1 rule-id 268434437
access-list CSM_FW_ACL_ line 10 advanced permit ip ifc DMZ host 10.10.10.1 ifc ISP-1 host 20.20.20.2 rule-id 268434437

access-list CSM_FW_ACL_ line 11 advanced permit ip ifc Inside object-group Source-hosts ifc ISP-1 object-group Destination-hosts rule-id
268434437
access-list CSM_FW_ACL_ line 11 advanced permit ip ifc Inside host 10.10.10.2 ifc ISP-1 host 20.20.20.1 rule-id 268434437
access-list CSM_FW_ACL_ line 11 advanced permit ip ifc Inside host 10.10.10.2 ifc ISP-1 host 20.20.20.2 rule-id 268434437
access-list CSM_FW_ACL_ line 11 advanced permit ip ifc Inside host 10.10.10.1 ifc ISP-1 host 20.20.20.1 rule-id 268434437
access-list CSM_FW_ACL_ line 11 advanced permit ip ifc Inside host 10.10.10.1 ifc ISP-1 host 20.20.20.2 rule-id 268434437
```

- Rule expansion with OGS enabled.



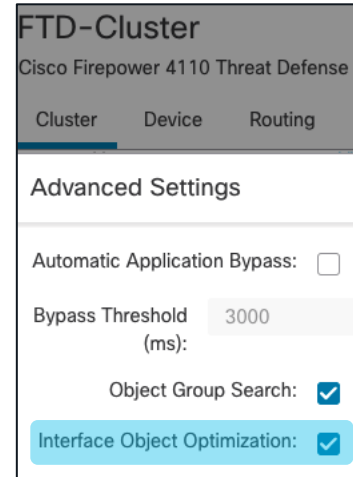
```
firepower# show access-list
access-list CSM_FW_ACL_ line 10 advanced permit ip ifc DMZ object-group Source-hosts ifc ISP-1 object-group Destination-hosts rule-id
268434437
access-list CSM_FW_ACL_ line 10 advanced permit ip ifc DMZ v4-object-group Source-hosts(2147483648) ifc ISP-1 v4-object-group
Destination-hosts(2147483649) rule-id 268434437
access-list CSM_FW_ACL_ line 11 advanced permit ip ifc Inside object-group Source-hosts ifc ISP-1 object-group Destination-hosts rule-
id 268434437
access-list CSM_FW_ACL_ line 11 advanced permit ip ifc Inside v4-object-group Source-hosts(2147483648) ifc ISP-1 v4-object-group
Destination-hosts(2147483649) rule-id 268434437
```



# Access Control Rule Optimization

## Interface Object Optimization (IOO)

- Interface object-group support on FTD 6.7+
  - Object-group CLI is enhanced to support interface type
- Interface Object-Group is supported for advanced Access-List
- Object Group Search is enhanced to support Interface Object Group





# Access Control Rule Optimization

## Interface Object Optimization (IOO)

- Rule expansion with IOO disabled.

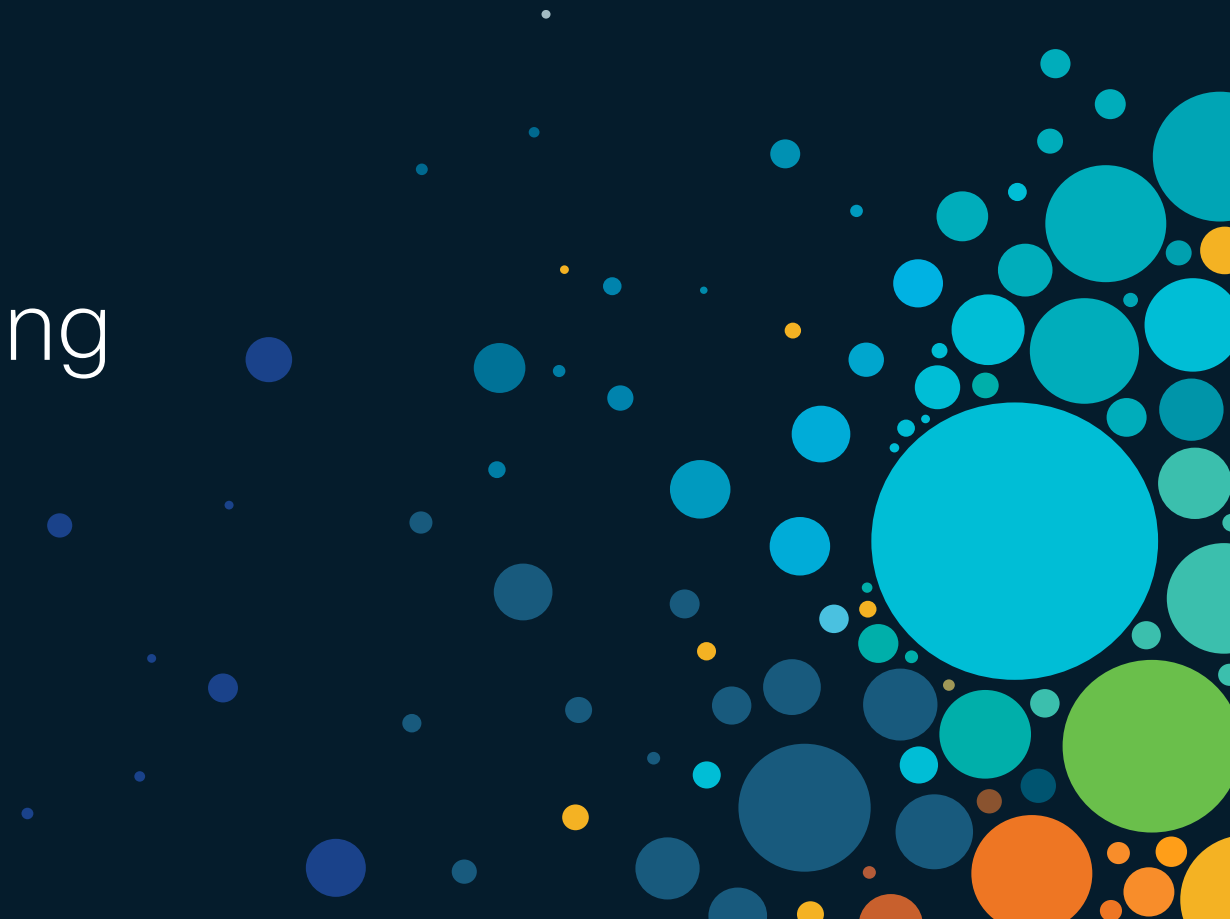
```
firepower# show access-list
access-list CSM_FW_ACL_line 10 advanced permit ip ifc DMZ object-group Source-hosts ifc ISP-1 object-group Destination-hosts rule-id
268434437
  access-list CSM_FW_ACL_line 10 advanced permit ip ifc DMZ v4-object-group Source-hosts(2147483648) ifc ISP-1 v4-object-group
Destination-hosts(2147483649) rule-id 268434437
access-list CSM_FW_ACL_line 11 advanced permit ip ifc Inside object-group Source-hosts ifc ISP-1 object-group Destination-hosts rule-
id 268434437
  access-list CSM_FW_ACL_line 11 advanced permit ip ifc Inside v4-object-group Source-hosts(2147483648) ifc ISP-1 v4-object-group
Destination-hosts(2147483649) rule-id 268434437
```



- Rule expansion with IOO enabled.

```
firepower# show access-list
access-list CSM_FW_ACL_line 10 advanced permit ip object-group-ifc InternalZones object-group Source-hosts object-group-ifc
ExternalZones object-group Destination-hosts rule-id 268434437
  access-list CSM_FW_ACL_line 10 advanced permit ip object-group-ifc igsz_00000_zsgi v4-object-group Source-hosts(2147483648) object-
group-ifc igsz 00001 zsgi v4-object-group Destination-hosts(2147483649) rule-id 268434437
```

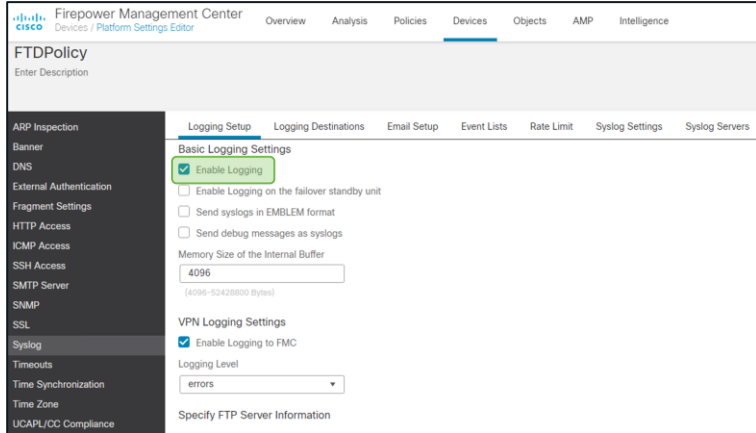
# Troubleshooting Tools



# Tools – Syslogs

- Syslogs remain the primary mechanism for recording connections **to** and **through** the firewall
- Should be the **first** troubleshooting tool to use for most issues
- Most syslogs in FTD are still generated from Lina:
  - Health of Lina resources and processes
  - Lina CPU, memory, block depletion
  - Failover events
  - NAT translation builds/teardowns

Note: Lina syslog config is defined under 'Platform Settings' in FMC



The screenshot shows the Cisco Firepower Management Center (FMC) interface for configuring a device's Syslog settings. The breadcrumb trail is "Firepower Management Center > Devices / Platform Settings Editor". The main heading is "FTDPolicy" with a sub-heading "Enter Description". The left sidebar lists various configuration categories, with "Syslog" selected. The main content area is titled "Logging Setup" and contains the following settings:

- Basic Logging Settings**
  - Enable Logging
  - Enable Logging on the failover standby unit
  - Send syslogs in EMBLEM format
  - Send debug messages as syslogs
  - Memory Size of the Internal Buffer: 4096 (4096-52428800 Bytes)
- VPN Logging Settings**
  - Enable Logging to FMC
  - Logging Level: errors
- Specify FTP Server Information

# Tools – Syslogs – FMC vs. CLI configuration



Cisco Secure  
Firewall YouTube  
Channel

- FMC screenshots and corresponding Lina CLI configuration:

1

Logging Setup Logging Destinations Email Setup

Basic Logging Settings

Enable Logging

Enable Logging on the failover standby unit

1

2

3

```
firepower# show run logging
logging enable
logging trap informational
logging host outside 10.1.0.1
```

2

Logging Setup Logging Destinations Email Setup Event Lists Rate Limit Syslog Settings Syslog Servers

Logging Destination Syslog from All Event Class Syslog from specific Event Class

Syslog Servers Filter on Severity: informational

3

Logging Setup Logging Destinations Email Setup Event Lists Rate Limit Syslog Settings Syslog Servers

Edit Syslog Server

Allow user traffic to pass when TCP s...

Message Queue Size(messages)\* 512

IP Address\* Syslog\_Server

Protocol  TCP  UDP

Port 514 (514 or 1025-65535)

Log Messages in Cisco EMBLEM format(UDP only)

Enable secure syslog.

Reachable By:

Device Management Interface (Applicable on FTD v6.3.0 and above)

Security Zones or Named Interface

Available Zones C

Q Search Add

Selected Zones/Interfaces

Outside

Interface Name Add

Cancel OK

Note: The syslog\_server object is defined as 10.1.0.1

# Tools – Syslogs – Connection Logging

- Lina connection logging and packet deny logs are **disabled** by default in FTD

CLI:

```
firepower# show run logging
...
no logging message 106015
no logging message 313001
no logging message 313008
no logging message 106023
no logging message 710003
no logging message 302015
no logging message 302014
no logging message 302013
no logging message 302019
no logging message 302017
no logging message 302016
no logging message 302021
no logging message 302020
```

Packet denials and  
ACL logging

UDP, TCP, GRE,  
and ICMP  
connections

FMC:

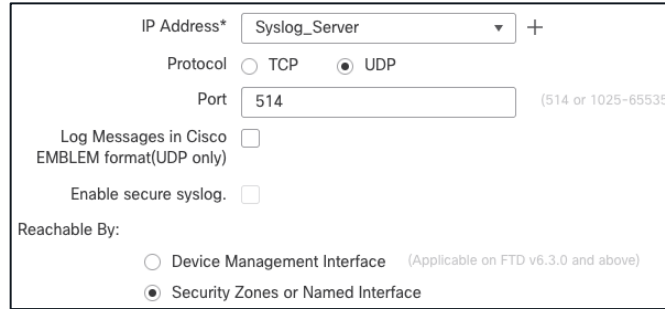
The screenshot shows the 'Syslog Settings' page in FMC. The 'Facility' is set to 'LOCAL4(20)'. The 'Enable Timestamp on Syslog Messages' checkbox is checked. The 'Timestamp Format' is set to 'Legacy (MMM dd yyyy HH:mm:ss)'. The 'Enable Syslog Device ID' and 'NetFlow Equivalent Syslogs' checkboxes are unchecked.

Syslog ID	Logging Level	Enabled
106015	(default)	•
106023	(default)	•
302013	(default)	•
302014	(default)	•
302015	(default)	•
302016	(default)	•
302017	(default)	•
302018	(default)	•
302020	(default)	•



# Tools – FTD Unified Syslogging

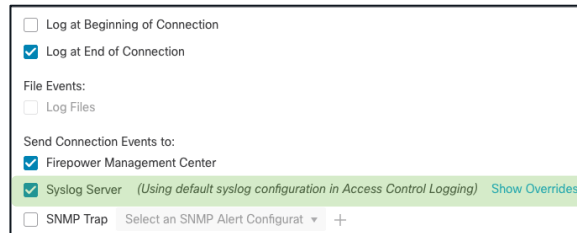
- In FTD 6.3 and later, syslogs can be generated from a single IP address (FTD management interface)



The screenshot shows the Syslog configuration interface for FTD. It includes the following fields and options:

- IP Address\*: Syslog\_Server (dropdown menu)
- Protocol:  TCP,  UDP
- Port: 514 (text input, with a note "(514 or 1025-65535)")
- Log Messages in Cisco EMBLEM format(UDP only):
- Enable secure syslog.:
- Reachable By:  Device Management Interface (Applicable on FTD v6.3.0 and above),  Security Zones or Named Interface

- %ASA- prefix changed to %FTD- and is also prepended to Snort logs
- Logging configuration in Platform Settings can be propagated to Access Control Policy



The screenshot shows the Platform Settings for logging. It includes the following options:

- Log at Beginning of Connection
- Log at End of Connection
- File Events:
  - Log Files
- Send Connection Events to:
  - Firepower Management Center
  - Syslog Server (Using default syslog configuration in Access Control Logging) [Show Overrides](#)
  - SNMP Trap (Select an SNMP Alert Configurat...)

# Tools – Syslogs – Snort vs. Lina

- Example: Logging at **beginning** AND **end** of connection AND **syslog** options for AC rule with Lina connection logging messages enabled in Syslog settings.

Date	Time	Priority	Hostname	Message
5/24/17	17:30:24	System4.Alert	10.1.1.79	May 24 21:30:22 FPR4100 SFIMS: Protocol: TCP, SrcIP: 10.1.1.20, OriginalClientIP: ::, DstIP: 172.18.124.145, SrcPort: 50072, DstPort: 21, TCPFlags: 0x0, DE: Primary Detection Engine (51a7d9fa-2943-11e7-80c4-bd73daa17015), Policy: 4120_Access_Policy, <b>ConnectType: End,</b> <b>AccessControlRuleName: Allow_Hosts,</b> <b>AccessControlRuleAction: Allow,</b> Username: No Authentication Required, Client: FTP client, ApplicationProtocol: FTP, InitiatorPackets: 6, ResponderPackets: 6, InitiatorBytes: 434, ResponderBytes: 462, DNSResponseType: No Error, Sinkhole: Unknown, URLCategory: Unknown, URLReputation: Risk unknown
5/24/17	17:30:17	System4.Alert	10.1.1.79	May 24 21:30:17 FPR4100 SFIMS: Protocol: TCP, SrcIP: 10.1.1.20, OriginalClientIP: ::, DstIP: 172.18.124.145, SrcPort: 50072, DstPort: 21, TCPFlags: 0x0, DE: Primary Detection Engine (51a7d9fa-2943-11e7-80c4-bd73daa17015), Policy: 4120_Access_Policy, <b>ConnectType: Start,</b> <b>AccessControlRuleName: Allow_Hosts,</b> <b>AccessControlRuleAction: Allow,</b> Username: No Authentication Required, InitiatorPackets: 2, ResponderPackets: 1, InitiatorBytes: 148, ResponderBytes: 78, DNSResponseType: No Error, Sinkhole: Unknown, URLCategory: Unknown, URLReputation: Risk unknown
5/24/17	17:30:24	Local4.I nfo	10.1.1.80	%ASA-6-302014: <b>Teardown TCP connection 14704 for inside:10.1.1.20/50072</b> to outside:172.18.124.145/21 duration 0:00:05 bytes 40 Flow closed by inspection
5/24/17	17:30:18	Local4.I nfo	10.1.1.80	%ASA-6-302013: <b>Built inbound TCP connection 14704 for inside:10.1.1.20/50072</b> (10.2.104.80/50072) to outside:172.18.124.145/21 (172.18.124.145/21)

Build

Snort Policy

Snort Action

Teardown

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

Levels	
0—Emergency	4—Warning
1—Alert	5—Notifications
2—Critical	6—Informational
3—Errors	7—Debugging

Add Syslog Settings

Syslog Id\* 111009

Logging Level errors

Enabled

Cancel OK

Devices Settings → Platform Settings → Syslog

# Logging – Common Issues

- **SNMP Trap as a logging destination** should only be used when you really have an SNMP server that you want to receive **all** syslogs
- **Logging to the console** should only be enabled **while actively troubleshooting** on the console
- **Logging on the standby unit** should only be used if you want to receive double the syslogs
- **Allow user traffic to pass when TCP syslog server is down** should nearly always be enabled with TCP syslogging

Logging Destination	SNMP Trap	
Event Class	Filter on Severity	emergencies

Logging Destination	Console	
Event Class	Filter on Severity	emergencies

Logging Setup	Logging Destinations
Basic Logging Settings	
<input checked="" type="checkbox"/>	Enable Logging
<input type="checkbox"/>	Enable Logging on the failover standby unit

Logging Setup	Logging Destinations	Email Setup	Event Lists	Rate Limit	Syslog Settings	Syslog Servers
<input checked="" type="checkbox"/>	Allow user traffic to pass when TCP syslog server is down (Recommended to be enabled)					

# Best Practices when issuing Debug Commands



Debugs should not be the first choice to troubleshoot a problem



Debugs can **negatively** impact the CPU complex and affect performance



Use conditional debugs, If Possible!



Know how much traffic of the matching type is passing through the firewall before enabling the respective debug

# Traffic Rates

Uptime statistics is useful to determine historical average packet size and rates:  
 $52128831 \text{ B/sec} / 39580 \text{ pkts/sec} = \sim 1317 \text{ B/packet}$

```
firepower# 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
```

One-minute average is useful to detect bursts and small packets:  
 $25190735 \text{ B/sec} / 144028 \text{ pkts/sec} = \sim 174 \text{ B/packet}$

# Xlate Table

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

```
firepower# 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

```
firepower# 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 (6.7 and below)



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

```
firepower# 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



# Detailed NAT Information (7.0 and above)



- In Firepower version 7.0, Section 0 was added to the NAT table for all implicit NAT rules for NLP applications (sftunnel, SSH, SNMP, HTTP, DNS)

```
firepower# show nat detail
Manual NAT Policies Implicit (Section 0)
1 (nlp_int_tap) to (Inside) source static nlp_server_ssh_0.0.0.0_intf4 interface destination
static 0_0.0.0.0_2 0_0.0.0.0_2 service tcp ssh ssh
  translate_hits = 0, untranslate_hits = 0
  Source - Origin: 169.254.1.2/32, Translated: 192.168.10.1/24
  Destination - Origin: 0.0.0.0/0, Translated: 0.0.0.0/0
  Service - Protocol: tcp Real: ssh Mapped: ssh
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 (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.

Untranslate hits indicate connections from mapped to real interfaces

Translate hits indicate connections from real to mapped interfaces

# Connection Table

```
firepower# show conn detail
2 in use, 7 most used
Inspect Snort:
  preserve-connection: 1 enabled, 0 in effect, 6 most enabled, 0 most in effect
Flags: A - awaiting responder ACK to SYN, a - awaiting initiator ACK to SYN,
      B - TCP probe for server certificate,
      b - TCP state-bypass or nailed,
      C - CTIQBE media, c - cluster centralized,
      D - DNS, d - dump, E - outside back connection, e - semi-distributed,
      F - initiator FIN, f - responder FIN,
      G - group, g - MGCP, H - H.323, h - H.225.0, I - initiator data,
      i - incomplete, J - GTP, j - GTP data, K - GTP t3-response
      k - Skinny media, L - decap tunnel, M - SMTP data, m - SIP media
      N - inspected by Snort (1 - preserve-connection enabled, 2 - preserve-connection in effect)
      n - GUP, O - responder data, o - offloaded,
      P - inside back connection, p - passenger flow
      q - SQL*Net data, R - initiator acknowledged FIN,
      R - UDP SUNRPC, r - responder acknowledged FIN,
      T - SIP, t - SIP transient, U - up,
      V - VPN orphan, v - M3UA W - WAAS,
      w - secondary domain backup,
      X - inspected by service module,
      x - per session, Y - director stub flow, y - backup stub flow,
      Z - Scansafe redirection, z - forwarding stub flow
```

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

Conn flags indicate current state

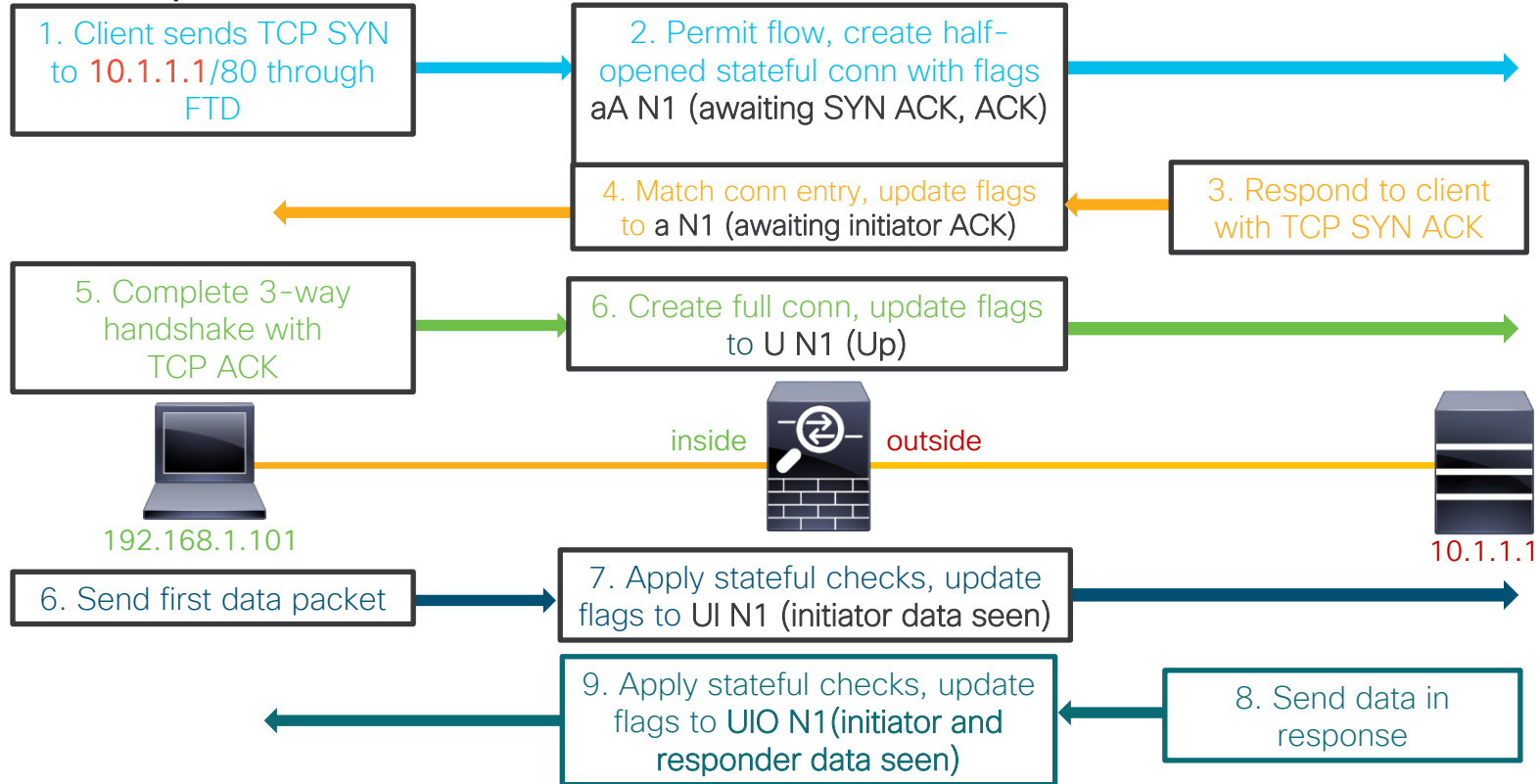
N flag shows if connection is sent to snort:  
N1: connection eligible for preserve-connection  
N2: preserve-connection enabled for the connection in question

Bidirectional byte count;

Used for connection debugging

```
TCP Inside: 192.168.45.130/39978 ISP1: 192.168.10.31/21,
  flags UxIO N1, idle 19s, uptime 24s, timeout 1h0m, bytes 728, xlate id 0x150406257f80
Initiator: 192.168.45.130, Responder: 192.168.10.31
Connection lookup keyid: 34422758
```

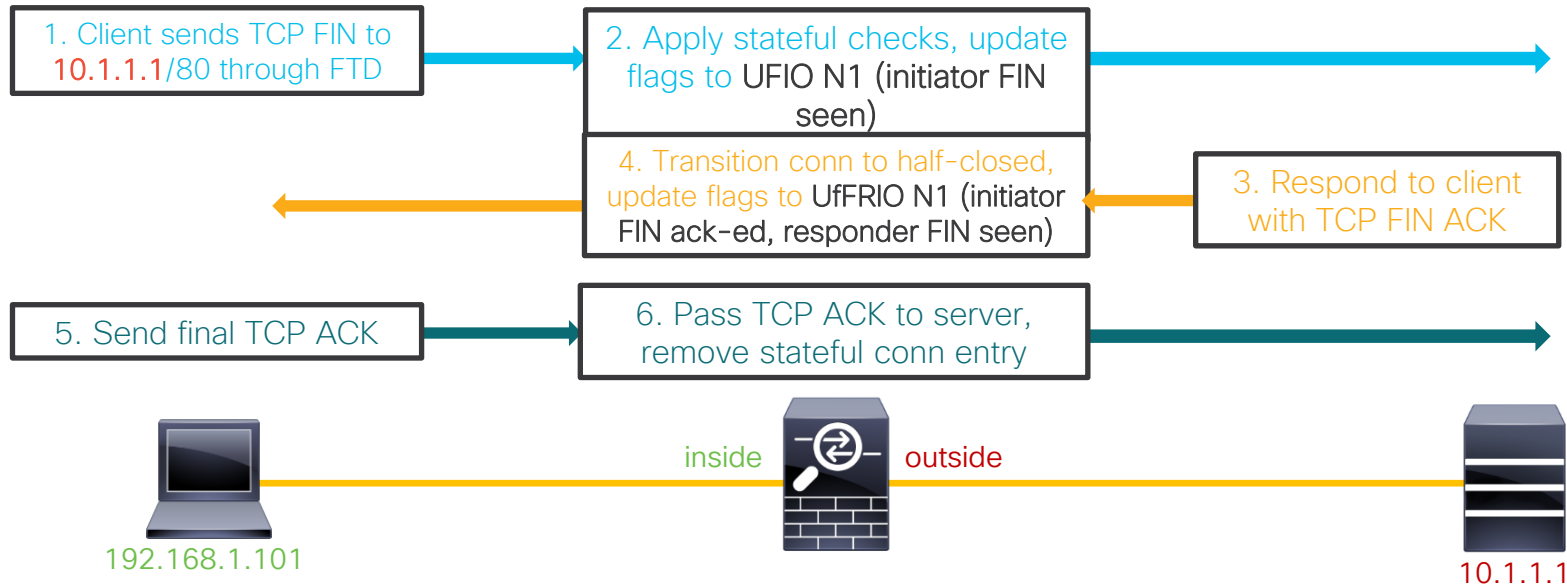
# Example: TCP Connection Establishment



TCP **outside** 10.1.1.1:80 **inside** 192.168.1.101:50141, idle 0:00:00, bytes 153, flags **UIO N1**

# Example: TCP Connection Termination

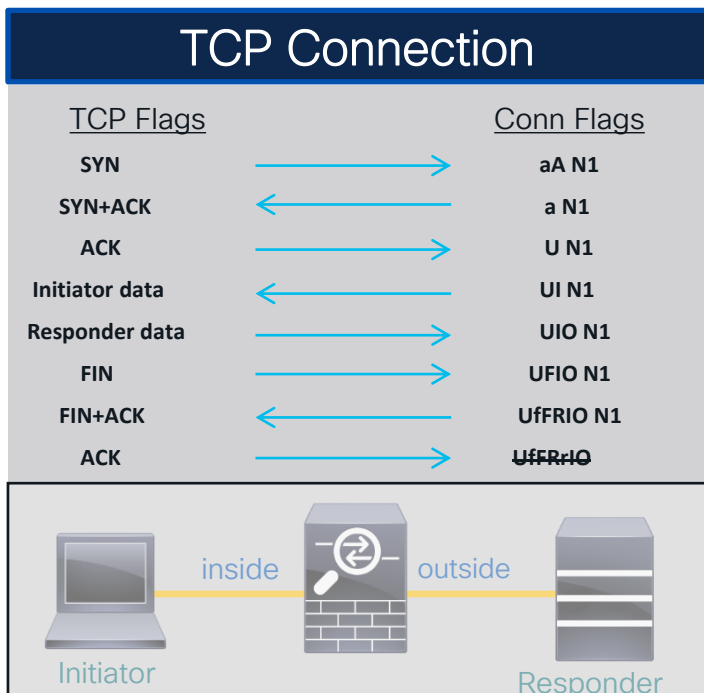
TCP **outside** 10.1.1.1:80 **inside** 192.168.1.101:50141, idle 0:00:00, bytes 153, flags **UIO N1**





For your reference

# TCP Connection Flags in FTD



a	Awaiting initiator ACK to SYN
A	Awaiting responder ACK to SYN
U	Up - 3way Handshake complete
I	Received Initiator Data
O	Received Responder Data
F	Received Initiator FIN
f	Received Responder FIN
R	Received Initiator ACK to FIN
N1	Inspected by Snort

# TCP Connection Termination Reasons

- If logging messages are enabled and a TCP flow was built through FTD, it will **always** log a teardown reason
- TCP teardown message is logged at level 6 (informational) by default
- For problems with abnormal connection termination, temporarily increase logging 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
```



For your  
reference

# 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





For your  
reference

# 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

Firepower 6.7.0.3 and Earlier

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

```
firepower# show local-host detail connection tcp 50 embryonic
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)
```

Can be added to show only half-open connections

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



# Local Host Table

FROM Firepower 7.0

- From Firepower version 7.0, Local-host entry will be created just when using: NAT, IDFW and Threat Detection

```
firepower# show local-host
Interface ISP2: 1 active, 1 maximum active
local host: <10.1.0.1>,
Interface DMZ: 0 active, 0 maximum active
Interface diagnostic: 0 active, 0 maximum active
Interface Inside: 1 active, 1 maximum active
local host: <192.168.45.130>,
  Xlate:
    TCP PAT from Inside:192.168.45.130/41076 to ISP1:192.168.10.37/41076 flags ri
idle 0:00:10 timeout 0:00:30
```

- To get information about total TCP/UDP connections per host, use ‘Show conn’ command

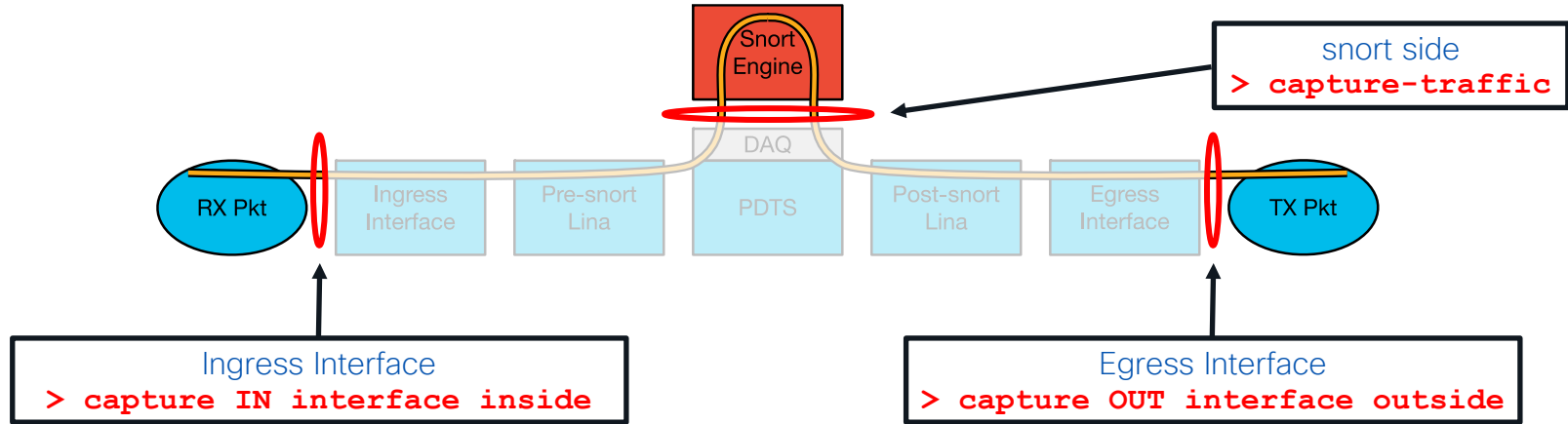
```
firepower# show conn address 192.168.45.130 | count TCP
Number of lines which match regexp = 2
```

# 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

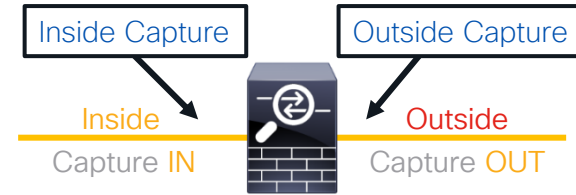
```
> show asp drop
Frame drop:
  Invalid encapsulation (invalid-encap)          10897
  Invalid tcp length (invalid-tcp-hdr-length)    9382
  Invalid udp length (invalid-udp-length)        10
  No valid adjacency (no-adjacency)             5594
  No route to host (no-route)                   1009
  Reverse-path verify failed (rpf-violated)      15
  Flow is denied by access rule (acl-drop)      25247101
  First TCP packet not SYN (tcp-not-syn)        36888
  Bad TCP Checksum (bad-tcp-cksum)             893
...
```

# Where Packets Are Captured in Packet Flow



- Ingress packets are captured **before** most packet processing
- Egress packets are captured **after** all processing
- “>capture-traffic” is a capture in snort which shows packets read from the DAQ

# Lina Packet Capture (CLI)



- Inline capability to record packets passing through FTD
- 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
  - Tcpcdump-like format for displaying captured packets on the box

```
firepower# capture OUT interface outside match ip any host 172.18.124.1
firepower# capture IN interface inside match ip any host 172.18.124.1
firepower# 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
firepower# no capture IN interface inside
firepower# no capture IN
```

Unlike ACL, match covers both directions of the flow

Removing the interface stops the capture but keeps contents in memory

Remember to remove the captures when done with troubleshooting

# Lina Packet Capture (CLI)

- Capture buffer maintained in RAM (512KB by default, 33 MB max)
  - Stops capturing when full by default, **circular** option available
- Default recorded packet length is 1518 bytes
- May elevate CPU utilization when applied under very high packet rates
- Copy captures off via FTP, SCP, or TFTP (example below)

```
firepower# capture OUT interface outside match ip any host 172.18.124.1  
firepower# copy /pcap capture:OUT tftp://10.10.1.1/capout.pcap
```

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

Configured capture name

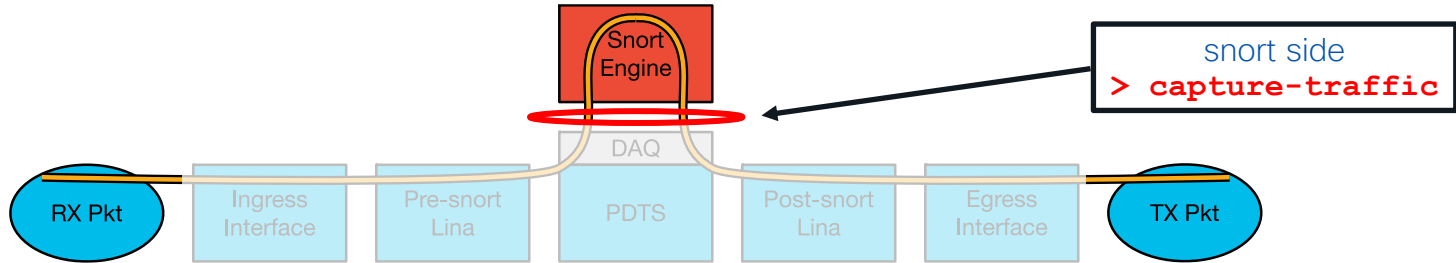
Save capture file under this name

# Packet Capture at time of Crash

- Allows use of a circular buffer to capture all traffic just before a crash occurs
- Very useful for troubleshooting traffic-related crashes

```
firepower# capture capin interface inside circular-buffer buffer 33000000
<<after forcing crash>>
firepower# show flash:
--#--  --length--  -----date/time-----  path
  109  198          Dec 09 2017 00:59:00  lina_phase1.log
<<output truncated>>
  110  1761873       Jan 22 2019 10:36:34  capin.pcap
  111  502025       Jan 22 2019 10:36:42  crashinfo_20190122_103635.UTC
```

# Snort-side captures with > capture-traffic



```
> capture-traffic
```

```
Please choose domain to capture traffic from:
```

- 0 - br1
- 1 - Router

```
Selection? 1
```

```
Please specify tcpdump options desired.  
(or enter '?' for a list of supported options)
```

```
Options: -n -s 0 -w SNORTCAP.pcap -c 1000 host 192.168.1.2 and port 80
```

tcpdump -c 1000  
Stop after 1000 packets

Standard BPF  
(Berkeley Packet Filter) Options

tcpdump -n  
Don't resolve hostnames

tcpdump -s 0  
Capture the whole packet

tcpdump -w FILE.pcap  
Write the capture to file

> capture-traffic  
PCAPs are written to:  
/ngfw/var/common/



# Capturing ASP drops

- Capture all frames dropped in the ASP

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

- Capture all frames with a specific drop reason

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

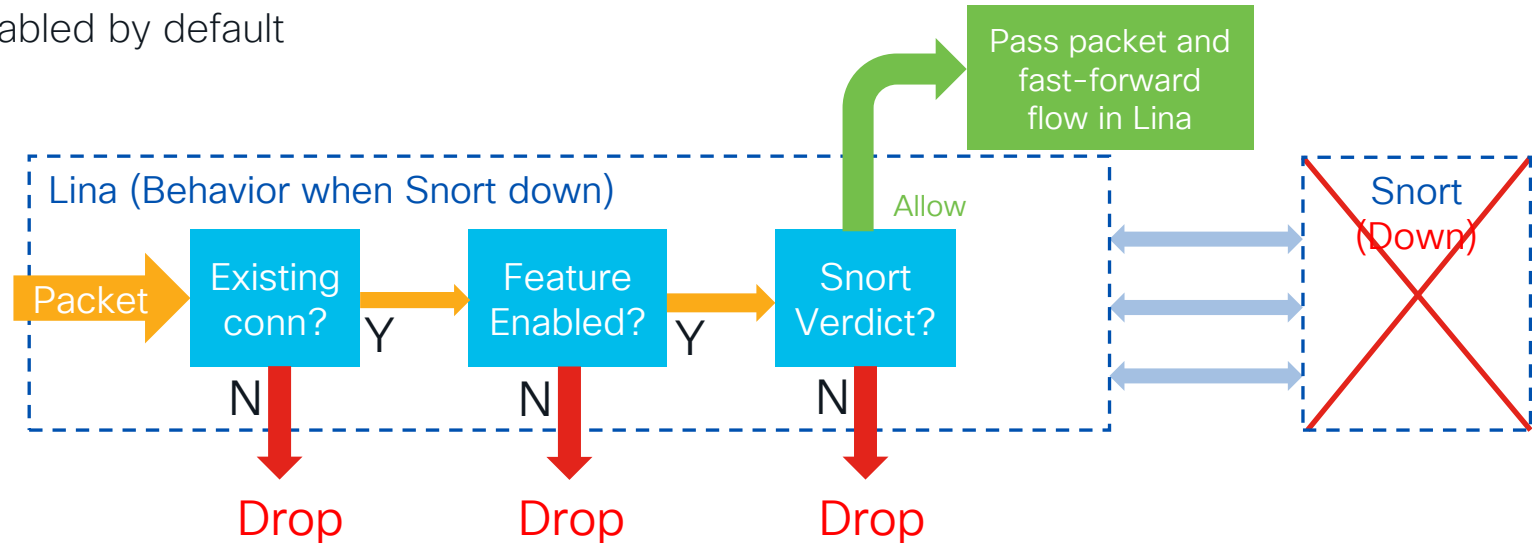
- ASP flow drops are non-atomic and cannot be captured

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

In FTD you can filter ASP drops using an inline match statement like a normal packet capture

# Snort Preserve-Connection

- Allows packets to pass while snort is down/restarting
- Flow must have reached an “Allow” verdict (AC policy)
- Added in 6.2.3
- Enabled by default



# Snort Preserve-Connection: Enable/Disable

## Show Current Setting

```
> show running-config snort  
snort preserve-connection
```

## Change Setting

```
> configure snort preserve-connection disable  
Building configuration...  
Cryptochecksum: 4fd6de40 7bf66af6 b1836604 04f8496d  
  
5745 bytes copied in 0.690 secs  
[OK]  
> show running-config snort  
no snort preserve-connection
```

# Snort Preserve-Connection: Troubleshooting

```
> show snort statistics
```

```
Packet Counters:
```

```
Passed Packets
```

```
Blocked Packets
```

```
Injected Packets
```

```
Packets bypassed (Snort Down)
```

```
Packets bypassed (Snort Busy)
```

```
62501
```

```
2339
```

```
5739
```

```
5678
```

```
0
```

Packets Preserved




```
[lines removed]
```

```
> show conn
```

```
0 in use, 231 most used
```

```
Inspect Snort:
```

```
preserve-connection: 14 enabled, 5 in effect, 215 most enabled, 40 most in effect
```



```
[lines removed]
```

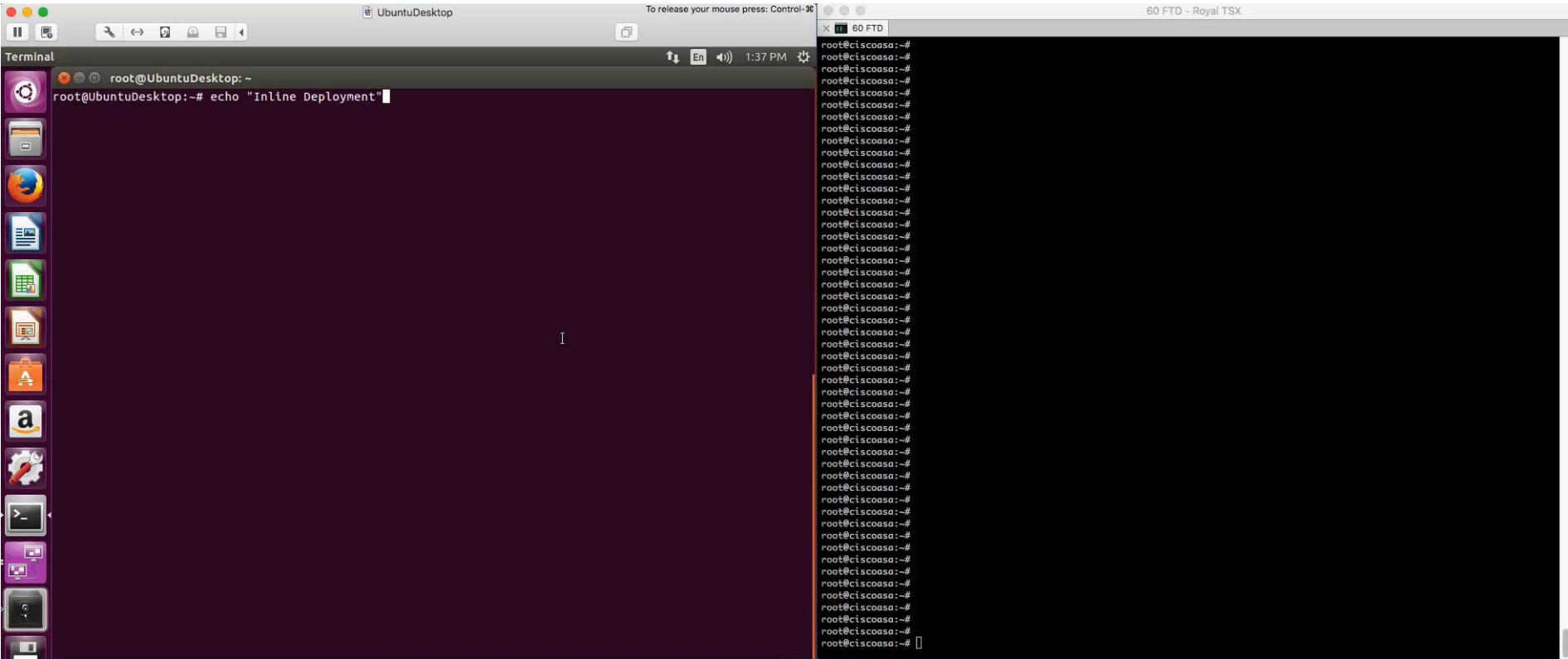
# How do I know if Snort Restarted or Reloaded?

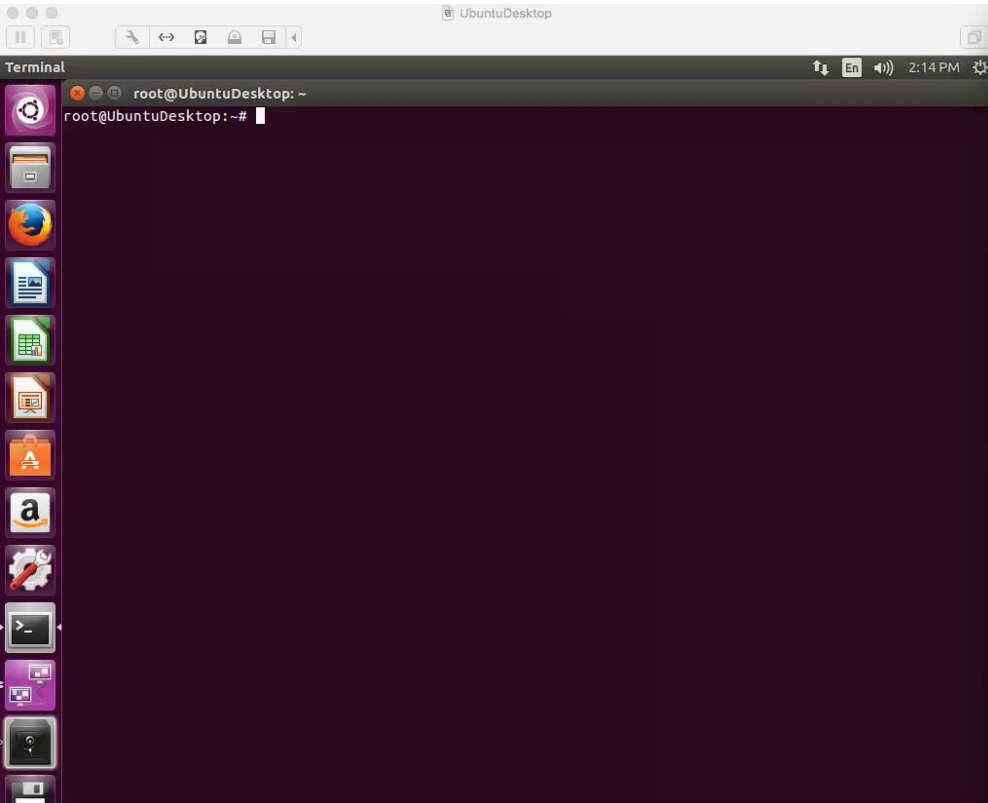
*This gets logged to /ngfw/var/log/messages*

```
root@ciscoasa:~# egrep "Initializing Snort|Reloading Snort" /ngfw/var/log/messages
Oct 9 11:53:07 ciscoasa SF-IMS[28379]:      --- Reloading Snort ---
Oct 9 11:53:07 ciscoasa SF-IMS[28380]:      --- Reloading Snort ---
Oct 9 11:59:18 ciscoasa SF-IMS[28379]:      }--- Reloading Snort ---
Oct 9 11:59:18 ciscoasa SF-IMS[28380]:      --- Reloading Snort ---
Oct 9 12:25:51 ciscoasa SF-IMS[28379]:      --- Reloading Snort ---
Oct 9 12:25:51 ciscoasa SF-IMS[28380]:      --- Reloading Snort ---
Oct 9 12:37:40 ciscoasa SF-IMS[28379]:      --- Reloading Snort ---
Oct 9 12:37:40 ciscoasa SF-IMS[28380]:      --- Reloading Snort ---
Oct 9 12:37:44 ciscoasa snort[4460]:      --- Initializing Snort ---
Jan 28 11:45:58 ciscoasa snort[4298]:      --- Initializing Snort ---
Jan 28 13:09:29 ciscoasa snort[13012]:      --- Initializing Snort ---
root@ciscoasa:~#
```

**RELOAD**

**RESTART**





```
60 FTD
Interface GigabitEthernet0/1 "Inside", is up, line protocol is up
Hardware is i82545EM rev01, 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 000c.2961.f795, MTU 1500
IP address 192.168.250.1, subnet mask 255.255.255.0
4331 packets input, 933137 bytes, 0 no buffer
Received 0 broadcasts, 0 runts, 0 giants
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
0 pause input, 0 resume input
0 L2 decode drops
19263 packets output, 11981755 bytes, 0 underruns
0 pause output, 0 resume output
0 output errors, 0 collisions, 1 interface resets
0 late collisions, 0 deferred
1 input reset drops, 0 output reset drops
input queue (blocks free curr/low): hardware (481/461)
output queue (blocks free curr/low): hardware (511/406)
Traffic Statistics for "Inside":
634 packets input, 53770 bytes
46 packets output, 3263 bytes
49 packets dropped
1 minute input rate 0 pkts/sec, 69 bytes/sec
1 minute output rate 0 pkts/sec, 5 bytes/sec
1 minute drop rate, 0 pkts/sec
5 minute input rate 0 pkts/sec, 81 bytes/sec
5 minute output rate 0 pkts/sec, 1 bytes/sec
5 minute drop rate, 0 pkts/sec
Interface GigabitEthernet0/2 "Passive_Replay_Recieve", is up, line protocol is up
Hardware is i82545EM rev01, 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 000c.2961.f79f, MTU 1500
IPS Interface-Mode: passive
IP address unassigned
0 packets input, 0 bytes, 0 no buffer
Received 0 broadcasts, 0 runts, 0 giants
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
0 pause input, 0 resume input
0 L2 decode drops
0 packets output, 0 bytes, 0 underruns
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 (511/511)
output queue (blocks free curr/low): hardware (511/511)
Traffic Statistics for "Passive_Replay_Recieve":
0 packets input, 0 bytes
0 packets output, 0 bytes
0 packets dropped
```

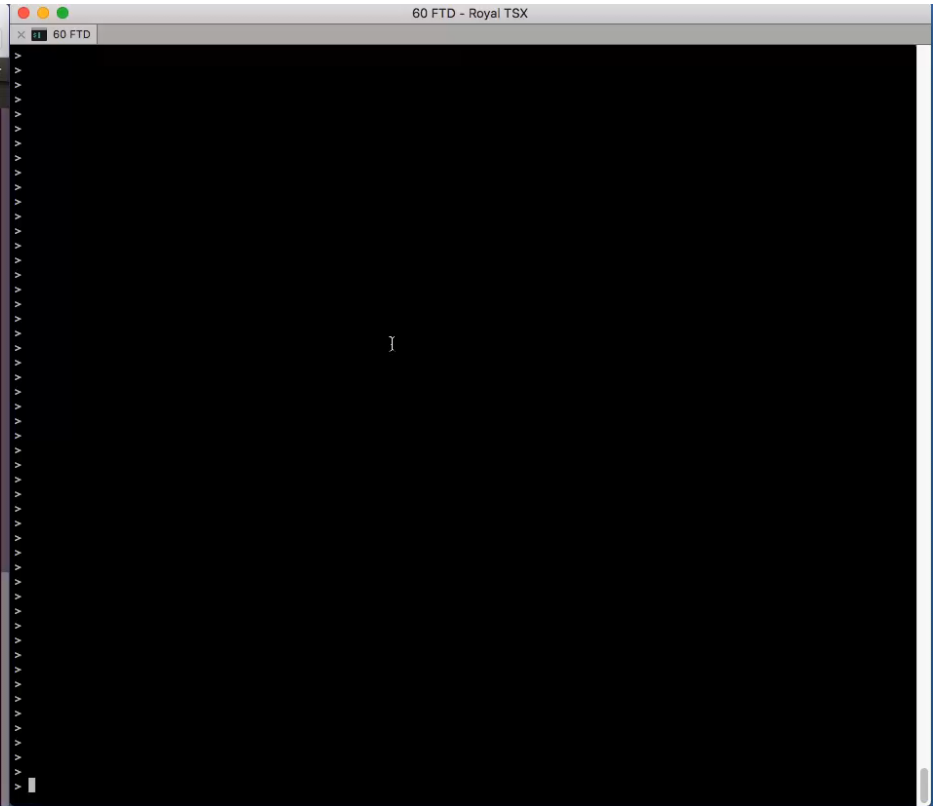
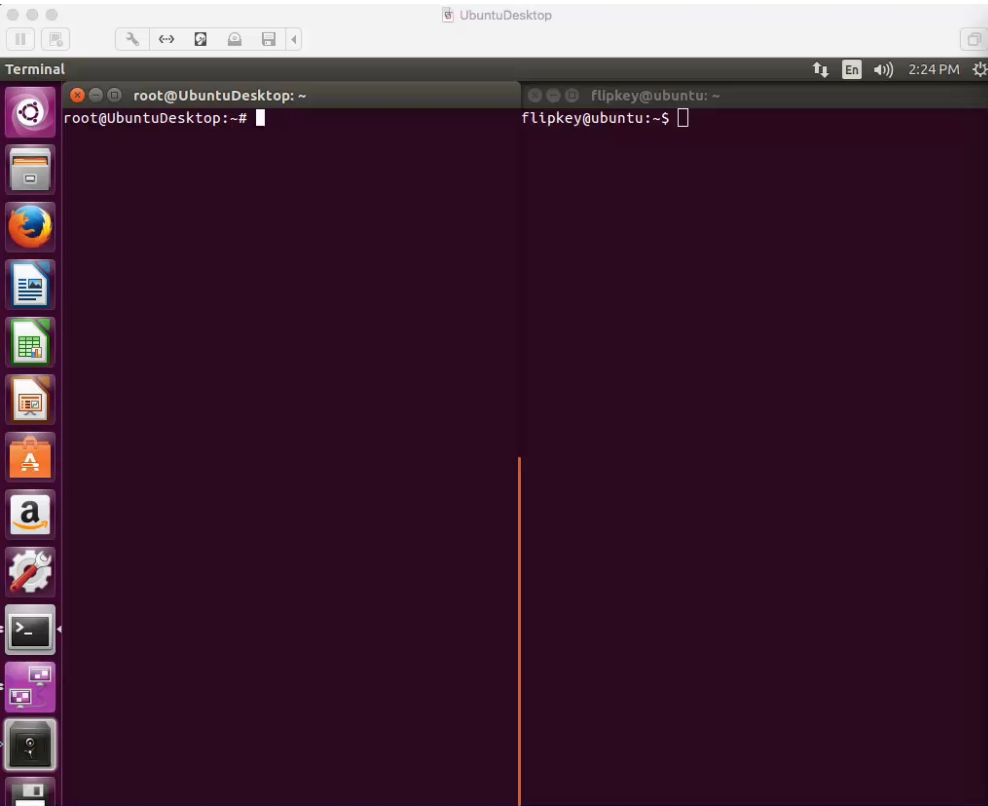
```
UbuntuDesktop
Terminal
root@UbuntuDesktop:~# echo routed
routed
root@UbuntuDesktop:~# ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=11.9 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=12.5 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=64 time=12.1 ms
64 bytes from 192.168.1.1: icmp_seq=4 ttl=64 time=11.9 ms
64 bytes from 192.168.1.1: icmp_seq=5 ttl=64 time=22.4 ms
64 bytes from 192.168.1.1: icmp_seq=6 ttl=64 time=22.5 ms
64 bytes from 192.168.1.1: icmp_seq=7 ttl=64 time=22.0 ms
64 bytes from 192.168.1.1: icmp_seq=8 ttl=64 time=11.7 ms
64 bytes from 192.168.1.1: icmp_seq=9 ttl=64 time=3.70 ms
64 bytes from 192.168.1.1: icmp_seq=10 ttl=64 time=12.3 ms
^C
--- 192.168.1.1 ping statistics ---
24 packets transmitted, 10 received, 58% packet loss, time 23116ms
rtt min/avg/max/mdev = 3.707/14.342/22.585/5.792 ms
root@UbuntuDesktop:~# ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=13 ttl=64 time=22.0 ms
64 bytes from 192.168.1.1: icmp_seq=14 ttl=64 time=32.7 ms
64 bytes from 192.168.1.1: icmp_seq=15 ttl=64 time=11.9 ms
64 bytes from 192.168.1.1: icmp_seq=16 ttl=64 time=11.9 ms
64 bytes from 192.168.1.1: icmp_seq=17 ttl=64 time=11.3 ms
64 bytes from 192.168.1.1: icmp_seq=18 ttl=64 time=11.7 ms
64 bytes from 192.168.1.1: icmp_seq=19 ttl=64 time=11.3 ms
64 bytes from 192.168.1.1: icmp_seq=20 ttl=64 time=13.0 ms
64 bytes from 192.168.1.1: icmp_seq=21 ttl=64 time=23.0 ms
64 bytes from 192.168.1.1: icmp_seq=22 ttl=64 time=11.9 ms
64 bytes from 192.168.1.1: icmp_seq=23 ttl=64 time=11.9 ms
^C
--- 192.168.1.1 ping statistics ---
23 packets transmitted, 11 received, 52% packet loss, time 22014ms
rtt min/avg/max/mdev = 11.347/15.747/32.748/6.749 ms
root@UbuntuDesktop:~#
```

```
60 FTD
0 packets output, 0 bytes
0 packets dropped
1 minute input rate 0 pkts/sec, 0 bytes/sec
1 minute output rate 0 pkts/sec, 0 bytes/sec
1 minute drop rate, 0 pkts/sec
5 minute input rate 0 pkts/sec, 0 bytes/sec
5 minute output rate 0 pkts/sec, 0 bytes/sec
5 minute drop rate, 0 pkts/sec
Interface Management0/0 "diagnostic", is up, line protocol is up
Hardware is en_vtun rev00, 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 000c-2961-f781, MTU 1500
IP address unassigned
360 packets input, 23530 bytes, 0 no buffer
Received 0 broadcasts, 0 runts, 0 giants
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
0 pause input, 0 resume input
0 L2 decode drops
0 packets output, 0 bytes, 0 underruns
0 pause output, 0 resume output
0 output errors, 0 collisions, 0 interface resets
0 late collisions, 0 deferred
0 input reset drops, 0 output reset drops
input queue (blocks free curr/low): hardware (0/0)
output queue (blocks free curr/low): hardware (0/0)
Traffic Statistics for "diagnostic":
360 packets input, 18364 bytes
0 packets output, 0 bytes
328 packets dropped
1 minute input rate 0 pkts/sec, 2 bytes/sec
1 minute output rate 0 pkts/sec, 0 bytes/sec
1 minute drop rate, 0 pkts/sec
5 minute input rate 0 pkts/sec, 0 bytes/sec
5 minute output rate 0 pkts/sec, 0 bytes/sec
5 minute drop rate, 0 pkts/sec
Management-only interface. Blocked 0 through-the-device packets

>
> configure snort preserve-connection disable
shell-init: error retrieving current directory: getcwd: cannot access parent directories: Permission denied
shell-init: error retrieving current directory: getcwd: cannot access parent directories: Permission denied
Building configuration...
Cryptochecksum: 86926298 3c18c63f f64246a7 7d06fe7d

8351 bytes copied in 0.120 secs
[OK]
> exit
root@ciscoasa:~# pmtool disablebytype snort
root@ciscoasa:~# pmtool enablebytype snort
root@ciscoasa:~# su adm
```





# Packet Tracer



# Packet Tracer

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

The diagram illustrates the configuration of Packet Tracer in a firewall. It shows two examples: a general configuration and an IPv6-specific configuration. Callouts explain the components of the configuration commands.

**Feature order and name** points to the `packet-tracer` command in the first example.

**Ingress interface** points to the `inside` interface in the first example.

**Packet information as it enters the ingress interface** points to the IP address and port information in the first example.

**Include detailed internal flow and policy structure information** points to the `detailed` keyword in the first example.

**IPv6 Example** points to the second configuration example.

```
firepower# packet-tracer input inside tcp 192.168.1.101 23121 172.16.171.125 23 detailed
Phase: 1
Type: CAPTURE
Subtype:
Result: ALLOW
Config:
Additional Information:
[...]

firepower# packet-tracer input inside tcp 2002:DB8:1:1::20 10000 2002:DB8:1:2::100 80
detailed
...
Result: ALLOW
Config:
Additional Information:
found next-hop 2002:db8:1:2::100 using egress ifc outside
```

# Sample Packet Tracer Output

```
firepower# 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
```

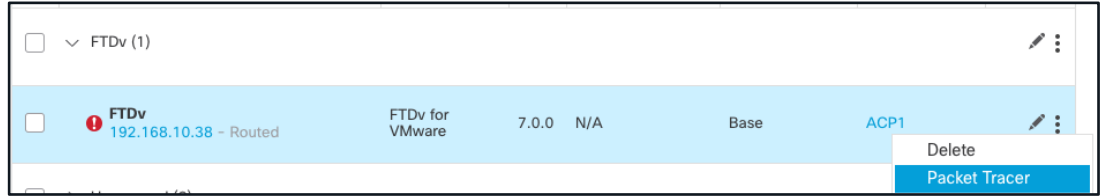
```
Untranslate 172.18.254.139/3389 to 192.168.103.221/3389
```

# Sample Packet Tracer Output (Cont'd)

```
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
```

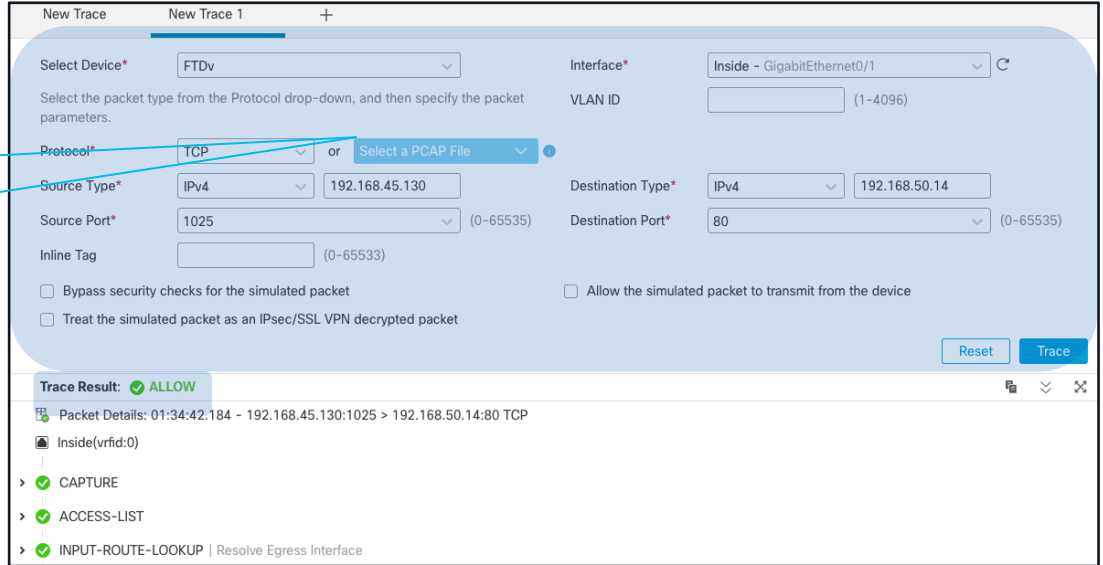
# Packet Tracer in FMC

1 Launch from Device Menu



2 Define Simulated packet

2a Select a PCAP File



3 Resulting action

# Packet Capture w/ Trace

- Enable packet tracer within an internal packet capture

```
firepower# capture IN interface inside trace trace-count 200 match tcp any any eq
```

Trace inbound  
packets only

Traced packet count per  
capture (1-1000, 50 by  
default)

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

```
firepower# 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

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

# Packet capture with trace (continued)

- Likely the **most used** datapath troubleshooting tool in the TAC
- You can now capture traffic post-decryption across a VPN tunnel w/ FTD as VPN endpoint:

```
firepower# capture OUT interface outside trace include-decrypted match tcp any any
```

New option captures packets that match the criteria after decryption

New packet-tracer option to allow egress of simulated packets

```
firepower# packet-tracer input inside tcp 10.1.1.20 10000 10.1.2.100 80 transmit detailed
firepower# sh cap capout
1 packet captured
  1: 12:08:30.837709      10.1.1.20.10000 > 10.1.2.100.80: S 1119191062:1119191062(0) win
```



# Firewall Engine Debug / System Support Trace

# Firewall Engine Debug (Snort)

- Shows Snort access control rule evaluation
- Indicates which rule a flow matches

```
> system support firewall-engine-debug
```

```
Please specify an IP protocol: tcp  
Please specify a client IP address: 192.168.1.2  
Please specify a client port: [redacted]  
Please specify a server IP address:  
Please specify a server port: 80
```

```
192.168.1.2-35948 > 172.16.2.10-80 6 AS 1 I 18 New session  
[lines_removed]
```

```
192.168.1.2-35948 > 172.16.2.10-80 6 AS 1 I 18 match rule order 2, 'Block Port HTTP  
Traffic', action Block
```

Common IP Header "Protocol" values:  
1 or "icmp"  
6 or "tcp"  
17 or "udp"

Leave a field blank for "any"

- Debug is written to messages log file  
`grep -i ngfwdbg /var/log/messages`

# System Support Trace (Snort)

> system support trace

- Debugs a flow in snort **per packet** (be careful!)
- Can optionally enable parallel firewall-engine-debug (recommended)
- Shows preprocessor impact (Network Analysis Policy) not shown in other outputs

```
> system support trace
```

```
[lines removed]
```

```
10.2.2.2-443 - 10.1.1.1-5623 6 Packet: TCP, ACK, seq 1448114540, ack 4072763547
10.2.2.2-443 - 10.1.1.1-5623 6 Firewall: allow rule, 'Allow_Inside_to_Outside', allow
10.2.2.2-443 - 10.1.1.1-5623 6 AppID: service HTTPS (1122), application Microsoft
(1423)
10.1.1.1-5623 > 10.2.2.2-443 6 Firewall: allow rule, 'Allow_Inside_to_Outside', allow
10.1.1.1-5623 > 10.2.2.2-443 6 NAP id 2, IPS id 0, Verdict PASS
```

NAP and IPS identifiers  
`/ngfw/var/sf/detection_engines/UUID/snort.conf`

Snort verdict sent to DAQ/PDTS

# Troubleshooting Protocol Preprocessors

Trace



Use system support trace to find blocks by preprocessors

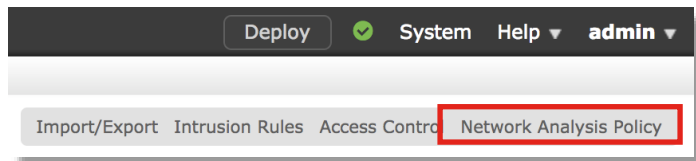
```
> system support trace
```

```
[omitted for brevity...]
```

```
172.16.111.226-51174 - 50.19.123.95-443 6 Packet: TCP, ACK, seq 3849839667, ack 1666843207
172.16.111.226-51174 - 50.19.123.95-443 6 Stream: TCP normalization error in timestamp, window, seq, ack,
fin, flags, or unexpected data, drop
172.16.111.226-51174 - 50.19.123.95-443 6 AppID: service unknown (0), application unknown (0)
172.16.111.226-51174 > 50.19.123.95-443 6 AS 4 I 0 Starting with minimum 3, 'block urls', and SrcZone first
with zones -1 -> -1, geo 0 -> 0, vlan 0, inline sgt tag: untagged, ISE sgt id: 0, svc 0, payload 0, client
0, misc 0, user 9999997, icmpType 0, icmpCode 0
172.16.111.226-51174 > 50.19.123.95-443 6 Firewall: starting rule matching, zone -1 -> -1, geo 0 -> 0, vlan
0, sgt 65535, user 9999997, icmpType 0, icmpCode 0
172.16.111.226-51174 > 50.19.123.95-443 6 AS 4 I 0 pending rule order 3, 'block urls', URL
172.16.111.226-51174 > 50.19.123.95-443 6 Firewall: pending rule-matching, 'block urls', pending URL
172.16.111.226-51174 > 50.19.123.95-443 6 Snort: processed decoder alerts or actions queue, drop
172.16.111.226-51174 > 50.19.123.95-443 6 IPS Event: gid 129, sid 14, drop
172.16.111.226-51174 > 50.19.123.95-443 6 NAP id 1, IPS id 0, Verdict BLOCK
172.16.111.226-51174 > 50.19.123.95-443 6 ==> Blocked by Stream
```

# Policies > Access Control > Intrusion

Disable Inline Mode



Edit or create a Network Analysis Policy

**Policy Information**

Name: My Custom NAP

Description:

Inline Mode:

Uncheck this box to disable Inline Mode

<u>Inline Result</u>	<u>Source IP</u>	<u>Destination IP</u>	<u>Source Port / ICMP Type</u>	<u>Destination Port / ICMP Code</u>	<u>Message</u>
	172.16.111.226	50.19.123.95	51177 / tcp	443 (https) / tcp	STREAM5_NO_TIMESTAMP (129:14:2)
↓	172.16.111.226	50.19.123.95	51174 / tcp	443 (https) / tcp	STREAM5_NO_TIMESTAMP (129:14:2)

Inline Mode disabled = No Inline Result

Inline Mode enabled = "Dropped" Inline Result

# Troubleshooting Protocol Preprocessors



**View preprocessors** → Settings

**Currently Enabled** {

- Back Orifice Detection
- Checksum Verification
- DCE/RPC Configuration
- DNS Configuration
- FTP and Telnet Configurat
- GTP Command Channel C
- HTTP Configuration
- Inline Normalization
- IP Defragmentation
- Packet Decoding
- SMTP Configuration
- SSH Configuration
- SSL Configuration
- Sun RPC Configuration
- TCP Stream Configuration
- UDP Stream Configuration

**Enabled with non-default settings** {

- Inline Normalization
- TCP Stream Configurat

**Enabled with default settings** {

- Back Orifice Detection
- Checksum Verification

**Settings** < Back

**Transport/Network Layer Preprocessors**

- Checksum Verification  Enabled  Disabled [Edit](#)
- Inline Normalization  Enabled  Disabled [Edit](#)
- IP Defragmentation  Enabled  Disabled [Edit](#)
- Packet Decoding  Enabled  Disabled [Edit](#)
- TCP Stream Configuration  Enabled  Disabled [Edit](#)
- UDP Stream Configuration  Enabled  Disabled [Edit](#)

**Specific Threat Detection**

- Back Orifice Detection  Enabled [Edit](#)

Filter:

GID:"129"

# Intrusion Policy



12 selected rules of 20

Policy

Rule State ▾ Event Filtering ▾ Dynamic State ▾ Alerting ▾ Comm

Generate Events

Drop and Generate Events

Disable

<input type="checkbox"/>	129	2	STREAM5_DATA_ON_SYN
<input checked="" type="checkbox"/>	129	3	STREAM5_DATA_ON_CLOSED
<input checked="" type="checkbox"/>	129	4	STREAM5_BAD_TIMESTAMP
<input type="checkbox"/>	129	5	STREAM5_BAD_SEGMENT
<input checked="" type="checkbox"/>	129	6	STREAM5_WINDOW_TOO_LARGE
<input type="checkbox"/>	129	7	STREAM5_EXCESSIVE_TCP_OVERLAPS
<input checked="" type="checkbox"/>	129	8	STREAM5_DATA_AFTER_RESET
<input type="checkbox"/>	129	9	STREAM5_SESSION_HIJACKED_CLIENT
<input type="checkbox"/>	129	10	STREAM5_SESSION_HIJACKED_SERVER
<input checked="" type="checkbox"/>	129	11	STREAM5_DATA_WITHOUT_FLAGS

## Policy Information ⚠

Settings

- Back Orifice Detection
- Checksum Verification
- DCE/RPC Configuration
- DNS Configuration
- FTP and Telnet Configuration
- GTP Command Channel Co
- HTTP Configuration
- Inline Normalization**
- IP Defragmentation
- Packet Decoding
- SMTP Configuration
- SSH Configuration
- SSL Configuration
- Sun RPC Configuration
- TCP Stream Configuration
- UDP Stream Configuration

## Inline Normalization

- Clear Urgent Pointer if URG=0
- Clear Urgent Pointer/URG on Empty Payload
- Clear URG if Urgent Pointer Is Not Set
- Normalize Urgent Pointer
- Normalize TCP Payload
- Remove Data on SYN
- Remove Data on RST
- Trim Data to Window
- Trim Data to MSS
- Block Unresolvable TCP Header Anomalies**

Network Analysis Policy



Still drops after setting to generate



Inline Result	Source IP	Destination IP	Source Port / ICMP Type	Destination Port / ICMP Code	Message
↓	172.16.111.226	50.19.123.95	51174 / tcp	443 (https) / tcp	STREAM5_NO_TIMESTAMP (129:14:2)
↓	172.16.111.226	50.19.123.95	51174 / tcp	443 (https) / tcp	STREAM5_NO_TIMESTAMP (129:14:2)

Inline Normalization



Check configuration guide for relative protocols/preprocessors:

### Block Unresolvable TCP Header Anomalies

When you enable this option, the system blocks anomalous TCP packets that, if normalized, would be invalid and likely would be blocked by the receiving host. For example, the system blocks any SYN packet transmitted subsequent to an established session.

The system also drops any packet that matches any of the following TCP stream preprocessor rules, regardless of whether the rules are enabled:

- 129:1
- 129:3
- 129:4
- 129:6
- 129:8
- 129:11
- 129:14 through 129:19

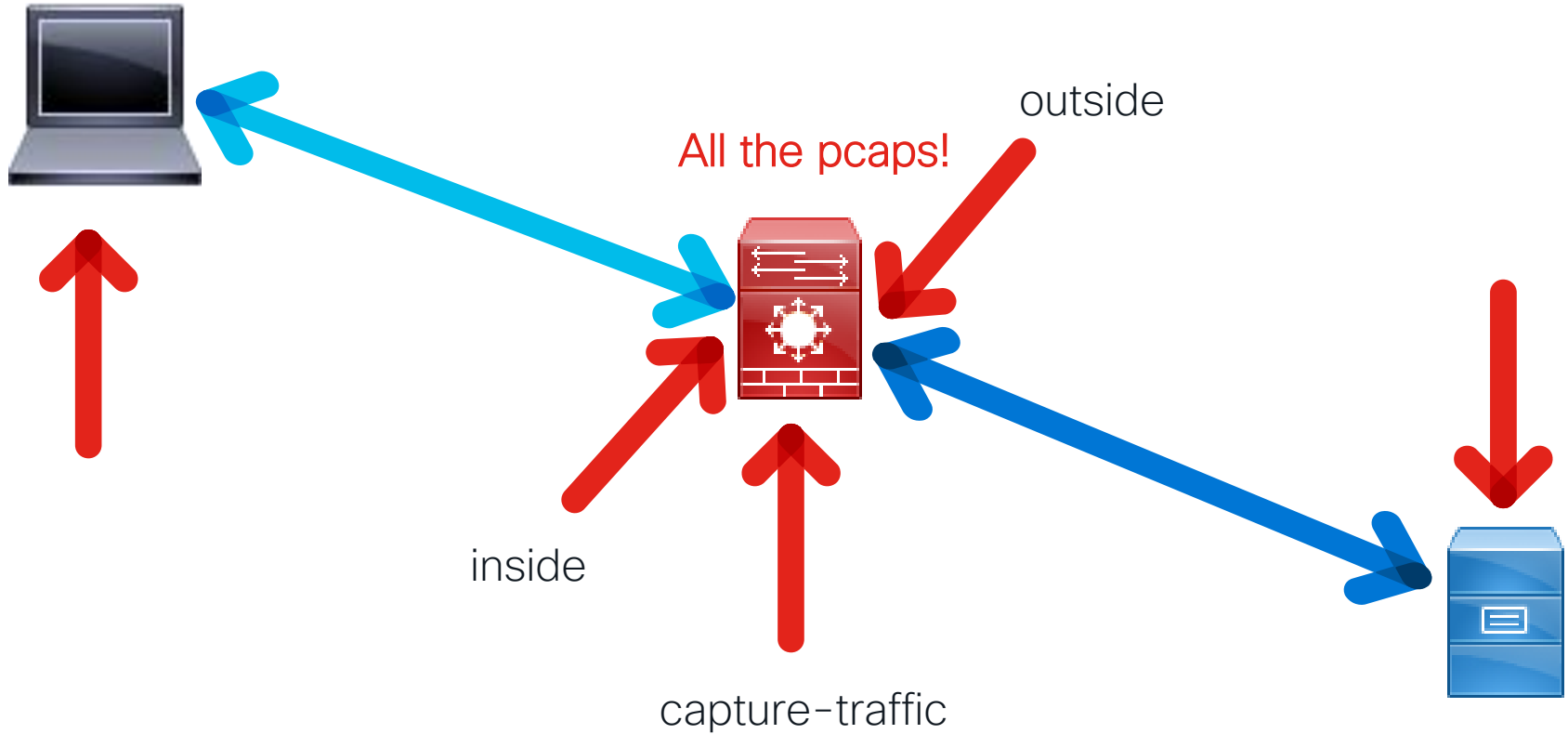
The Total Blocked Packets performance graph tracks the number of packets blocked in inline deployments and, in passive deployments and inline deployments in tap mode, the number that would have been blocked in an inline deployment.

Config guides: <http://www.cisco.com/c/en/us/support/security/defense-center/products-installation-and-configuration-guides-list.html>



# Packet Captures for SSL Decryption

# Pcaps



# Full handshake (Wireshark view)

Client Hello

443 → 55401 [ACK] Seq=1 Ack=206 Win=65535 Len=0

Server Hello

Certificate

55401 → 443 [ACK] Seq=206 Ack=1817 Win=64860 Len=0

Server Hello Done

Client Key Exchange, Change Cipher Spec, Encrypted Han...

Change Cipher Spec

# Pcap investigation: Client Hello

- Identify Handshake
- Session ID

```
▼ Secure Sockets Layer
  ▼ TLSv1.2 Record Layer: Handshake Protocol: Client Hello
    Content Type: Handshake (22)
    Version: TLS 1.0 (0x0301)
    Length: 200
  ▼ Handshake Protocol: Client Hello
    Handshake Type: Client Hello (1)
    Length: 196
    Version: TLS 1.2 (0x0303)
    > Random
    Session ID Length: 0
    Cipher Suites Length: 28
    > Cipher Suites (14 suites)
```

# Pcap investigation: Client Hello (continued)

- Session ID
- Server Name
- Known problems
- Potential problems

```
Version: TLS 1.2 (0x0303)
> Random
> Session ID Length: 0
  Cipher Suites Length: 28
> Cipher Suites (14 suites)
  Compression Methods Length: 1
> Compression Methods (1 method)
  Extensions Length: 127
> Extension: Unknown 23130
> Extension: renegotiation_info
> Extension: server_name
> Extension: Extended Master Secret
> Extension: SessionTicket TLS
> Extension: signature_algorithms
> Extension: status_request
> Extension: signed_certificate_timestamp
> Extension: Application Layer Protocol Negotiation
> Extension: channel_id
> Extension: ec_point_formats
> Extension: elliptic_curves
> Extension: Unknown 39578
```

# Pcap investigation: Server Hello

- Identify Handshake
- Session ID

```
▼ Secure Sockets Layer
  ▼ TLSv1.2 Record Layer: Handshake Protocol: Server Hello
    Content Type: Handshake (22)
    Version: TLS 1.2 (0x0303)
    Length: 81
  ▼ Handshake Protocol: Server Hello
    Handshake Type: Server Hello (2)
    Length: 77
    Version: TLS 1.2 (0x0303)
    > Random
    Session ID Length: 32
    Session ID: cdc9863a507daa0f1470ca0e19a4b3771a6a3ecf0ff3121d...
    Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)
    Compression Method: null (0)
    Extensions Length: 5
    > Extension: renegotiation_info
```

# Pcap investigation: Certificate

- Length
- Issuer

```

  v Certificates (1718 bytes)
    Certificate Length: 1715
  v Certificate: 308206af30820497a0030201020208150130034f311
    v signedCertificate
      version: v3 (2)
      serialNumber: 1513543740544848183
      > signature (sha256WithRSAEncryption)
      v issuer: rdnSequence (0)
        > rdnSequence: 6 items (id-at-commonName=Iseeyou)
        > validity
        > subject: rdnSequence (0)
        > subjectPublicKeyInfo
        > extensions: 5 items

```

tcp.stream eq 8

No.	Time	Source	Destination	Protocol	Length	Info	SRC PRT	New Column
1775	10.260709	192.168.1.200	172.217.8.10	TCP	66	59117 → 443 [SYN] Seq=0 Win=8192 Len=...	59117	443
1776	10.315668	172.217.8.10	192.168.1.200	TCP	66	443 → 59117 [SYN, ACK] Seq=0 Ack=1 W...	443	59117
1777	10.316186	192.168.1.200	172.217.8.10	TCP	54	59117 → 443 [ACK] Seq=1 Ack=1 Win=662...	59117	443
1778	10.318029	192.168.1.200	172.217.8.10	TLSv1.2	243	Client Hello	59117	443
1779	10.372759	172.217.8.10	192.168.1.200	TLSv1.2	373	Server Hello	443	59117
1780	10.373985	172.217.8.10	192.168.1.200	TCP	1514	[TCP segment of a reassembled PDU]	443	59117
1781	10.374095	172.217.8.10	192.168.1.200	TCP	1514	[TCP segment of a reassembled PDU]	443	59117
1782	10.374187	172.217.8.10	192.168.1.200	TLSv1.2	313	Certificate	443	59117
1783	10.374228	172.217.8.10	192.168.1.200	TLSv1.2	392	Server Key Exchange	443	59117
1784	10.374273	172.217.8.10	192.168.1.200	TLSv1.2	63	Server Hello Done	443	59117
1785	10.374302	192.168.1.200	172.217.8.10	TCP	54	443 → 59117 [RST] Seq=3837 Win=83884...	443	59117
1786	10.374322	192.168.1.200	172.217.8.10	TCP	54	59117 → 443 [RST] Seq=190 Win=262140 ...	59117	443

Decryption fails



Frame 1784: 63 bytes on wire (504 bits), 63 bytes captured (504 bits)  
 Ethernet II, Src: Vmware\_22:01:06 (00:0c:29:22:01:06), Dst: Vmware\_16:ac:87 (00:0c:29:22:01:06)  
 Internet Protocol Version 4, Src: 172.217.8.10, Dst: 192.168.1.200  
 Transmission Control Protocol, Src Port: 443 (443), Dst Port: 59117 (59117), Seq: 59117, Len: 63  
 Security Sockets Layer  
 TLSv1.2 Record Layer: Handshake Protocol: Server Hello Done  
 Content Type: Handshake (22)  
 Version: TLS 1.2 (0x0303)  
 Length: 4  
 Handshake Protocol: Server Hello Done  
 Handshake Type: Server Hello Done (14)  
 Length: 0



tcp.stream eq 10

No.	Time	Source	Destination	Protocol	Length	Info	SRC PRT	New Column
56	8.311365	192.168.1.200	98.138.199.240	TCP	66	59113 → 443 [SYN] Seq=0 Win=8192 Len=...	59113	443
57	8.383913	98.138.199.240	192.168.1.200	TCP	66	443 → 59113 [SYN, ACK] Seq=0 Ack=1 W...	443	59113
59	8.385263	192.168.1.200	98.138.199.240	TCP	54	59113 → 443 [ACK] Seq=1 Ack=1 Win=662...	59113	443
60	8.399074	192.168.1.200	98.138.199.240	TLSv1.2	239	Client Hello	59113	443
61	8.408633	98.138.199.240	192.168.1.200	TLSv1.2	124	Server Hello	443	59113
62	8.520860	98.138.199.240	192.168.1.200	TCP	1514	[TCP segment of a reassembled PDU]	443	59113
63	8.520986	98.138.199.240	192.168.1.200	TCP	1514	[TCP segment of a reassembled PDU]	443	59113
64	8.521085	98.138.199.240	192.168.1.200	TLSv1.2	433	Certificate	443	59113
65	8.521132	98.138.199.240	192.168.1.200	TLSv1.2	392	Server Key Exchange	443	59113
66	8.521251	98.138.199.240	192.168.1.200	TLSv1.2	63	Server Hello Done	443	59113
68	8.527142	192.168.1.200	98.138.199.240	TLSv1.2	129	Client Key Exchange	59113	443
69	8.613024	192.168.1.200	98.138.199.240	HTTP	809	POST /comet HTTP/1.1 (application/js...	59113	443
89	8.734453	98.138.199.240	192.168.1.200	TCP	294	[TCP segment of a reassembled PDU]	443	59113
90	8.734703	98.138.199.240	192.168.1.200	TCP	59	[TCP segment of a reassembled PDU]	443	59113
91	8.734878	98.138.199.240	192.168.1.200	TCP	1135	[TCP segment of a reassembled PDU]	443	59113
95	8.735212	98.138.199.240	192.168.1.200	HTTP	61	HTTP/1.1 200 OK (application/json)	443	59113
176	20.281141	192.168.1.200	98.138.199.240	HTTP	809	POST /comet HTTP/1.1 (application/js...	59113	443
177	20.429652	98.138.199.240	192.168.1.200	TCP	294	[TCP segment of a reassembled PDU]	443	59113

Frame 69: 809 bytes on wire (6472 bits), 809 bytes captured (6472 bits)  
 Ethernet II, Src: Vmware\_16:ac:87 (00:0c:29:16:ac:87), Dst: Vmware\_22:01:06 (00:0c:29:22:01:06)  
 Internet Protocol Version 4, Src: 192.168.1.200, Dst: 98.138.199.240  
 Transmission Control Protocol, Src Port: 59113 (59113), Dst Port: 443 (443), Seq: 261, Ack: 3717, Len: 755  
 Hypertext Transfer Protocol  
 [Expert Info (Warn/Security): Unencrypted HTTP protocol detected over encrypted port, could indicate a dangerous misconfiguration.]  
 [Unencrypted HTTP protocol detected over encrypted port, could indicate a dangerous misconfiguration.]  
 [Severity level: Warn]  
 [Group: Security]  
 POST /comet HTTP/1.1\r\n  
 [Expert Info (Chat/Sequence): POST /comet HTTP/1.1\r\n]  
 [POST /comet HTTP/1.1\r\n]  
 [Severity level: Chat]  
 [Group: Sequence]

Decryption succeeds



# Client Hello Modification

## Before

No.	Time	Source	Src Port	Destination
314	3.970446	192.168.1.200	63232	172.217.5
324	4.003952	192.168.1.200	63232	172.217.5
325	4.004944	192.168.1.200	63232	172.217.5
332	4.062085	192.168.1.200	63232	172.217.5
336	4.067959	192.168.1.200	63232	172.217.5
338	4.072598	192.168.1.200	63232	172.217.5
339	4.081448	192.168.1.200	63232	172.217.5
340	4.081463	192.168.1.200	63232	172.217.5

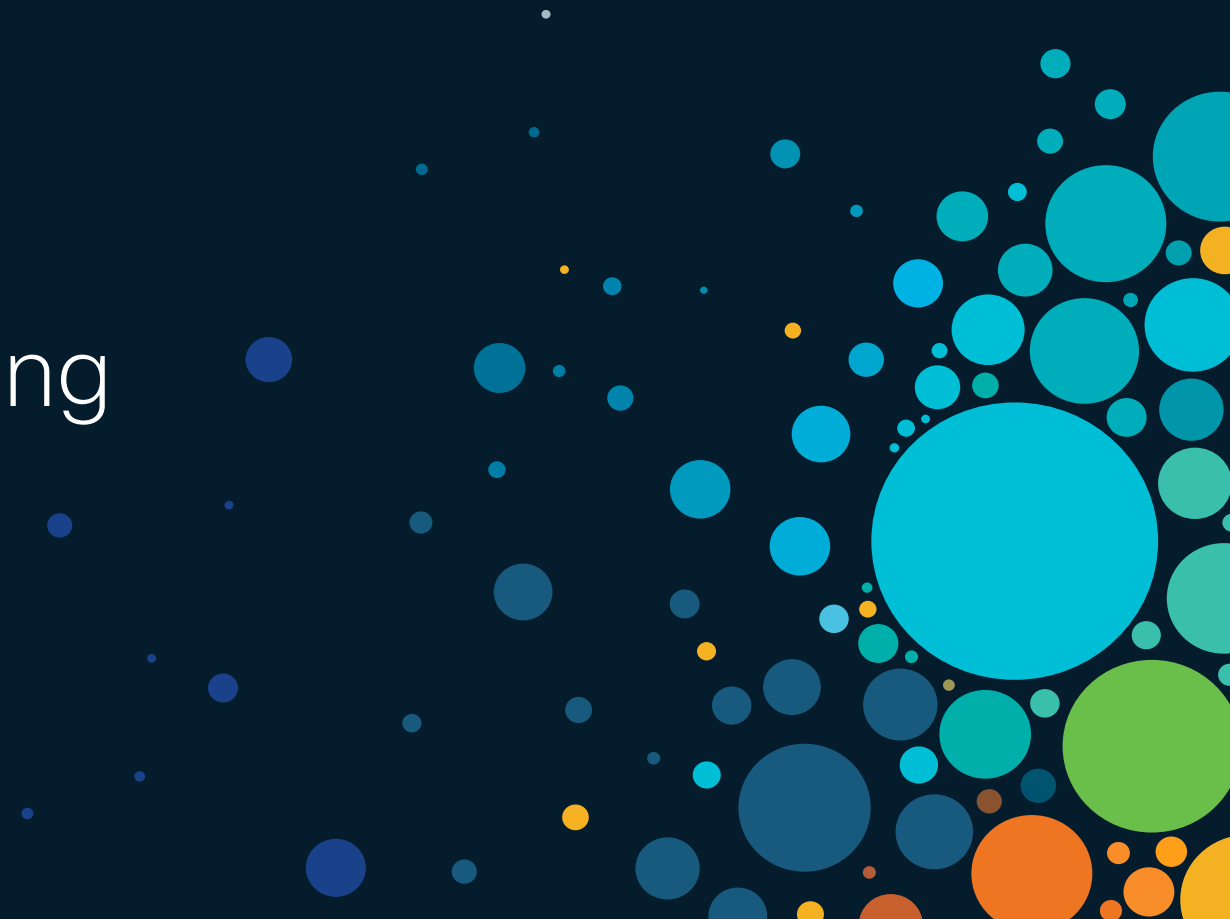
## After

No.	Time	Source	Src Port	Destination
707	12.608129	192.168.1.200	63232	172.217.5.226
717	12.634388	192.168.1.200	63232	172.217.5.226
718	12.636387	192.168.1.200	63232	172.217.5.226
730	12.672869	192.168.1.200	63232	172.217.5.226
734	12.697358	192.168.1.200	63232	172.217.5.226
738	12.711685	192.168.1.200	63232	172.217.5.226
739	12.712021	192.168.1.200	63232	172.217.5.226
740	12.712097	192.168.1.200	63232	172.217.5.226

- Handshake Protocol: Client Hello
  - Handshake Type: Client Hello (1)
    - Length: 208
    - Version: TLS 1.2 (0x0303)
      - Random
      - Session ID Length: 0
      - Cipher Suites Length: 28
      - Cipher Suites (14 suites)
      - Compression Methods Length: 1
      - Compression Methods (1 method)
      - Extensions Length: 139
        - Extension: Unknown 6682
        - Extension: renegotiation\_info
        - Extension: server\_name
        - Extension: Extended Master Secret
        - Extension: SessionTicket TLS
        - Extension: signature\_algorithms
        - Extension: status\_request
        - Extension: signed\_certificate\_timestamp
        - Extension: Application Layer Protocol Negotiation
        - Extension: channel\_id
        - Extension: ec\_point\_formats
        - Extension: elliptic\_curves
        - Extension: Unknown 56026

- Version: TLS 1.0 (0x0301)
  - Length: 194
- Handshake Protocol: Client Hello
  - Handshake Type: Client Hello (1)
    - Length: 190
    - Version: TLS 1.2 (0x0303)
      - Random
      - Session ID Length: 0
      - Cipher Suites Length: 22
      - Cipher Suites (11 suites)
      - Compression Methods Length: 1
      - Compression Methods (1 method)
      - Extensions Length: 127
        - Extension: Unknown 6682
        - Extension: renegotiation\_info
        - Extension: server\_name
        - Extension: SessionTicket TLS
        - Extension: signature\_algorithms
        - Extension: status\_request
        - Extension: signed\_certificate\_timestamp
        - Extension: Application Layer Protocol Negotiation
        - Extension: ec\_point\_formats
        - Extension: elliptic\_curves
        - Extension: Unknown 56026

# Identity Troubleshooting Tools



# Firewall engine debug

Firewall Engine Debug is the right tool to identify what is happening within the Access Control Policy

```
> system support firewall-engine-debug
```

```
Please specify an IP protocol: tcp
Please specify a client IP address: 172.16.1.2
Please specify a client port:
Please specify a server IP address: 192.168.0.10
Please specify a server port: 8081
```

```
Monitoring firewall engine debug messages
```

```
172.16.1.2-54255 > 192.168.0.10-8081 6 AS 1 I 0 New session
172.16.1.2-54255 > 192.168.0.10-8081 6 AS 1 I 0 Starting with minimum 4, 'Allow_Group2', and
IPProto first with zones 1 -> 2, geo 0 -> 0, vlan 0, inline sgt tag: untagged, ISE sgt id: 0,
svc 0, payload 0, client 0, misc 0, user 1, icmpType 0, icmpCode 0
172.16.1.2-54255 > 192.168.0.10-8081 6 AS 1 I 0 rule order 4, 'Allow_Group2', did not match
group 2
172.16.1.2-54255 > 192.168.0.10-8081 6 AS 1 I 0 no match rule order 4, 'Allow_Group2', user
1, realm 2
172.16.1.2-54255 > 192.168.0.10-8081 6 AS 1 I 0 match rule order 5, id 268434432 action Allow
172.16.1.2-54255 > 192.168.0.10-8081 6 AS 1 I 0 allow action
```

## ID of currently mapped user:

```
1 - 999999X = Downloaded User
9999995 = Pending User
9999996 = Guest
9999997 = No Auth Required
9999998 = Failed Authentication
9999999 = Unknown
```

# Identity-debug

The Identity-debug tool allows the user to troubleshoot the Identity Policy.

```
> system support identity-debug
```

```
Please specify an IP protocol: tcp
```

```
Please specify a client IP address: 172.16.1.2
```

```
Please specify a client port:
```

```
Please specify a server IP address: 192.168.0.10
```

```
Please specify a server port: 8081
```

```
Monitoring identity debug messages
```

```
172.16.1.2-43490 > 192.168.0.10-8081 6 AS 1 I 0 Starting authentication (sfAuthCheckRules params)
with zones 2 -> 3, port 43490 -> 8081, geo 16429296 -> 16429314
```

```
172.16.1.2-43490 > 192.168.0.10-8081 6 AS 1 I 0 Starting Auth SrcZone first with zones 2 -> 3, geo
2 -> 3, vlan 0
```

```
172.16.1.2-43490 > 192.168.0.10-8081 6 AS 1 I 0 Matched rule order 1, id 1, authRealmId 2, AD
Domain fire.int
```

```
172.16.1.2-43490 > 192.168.0.10-8081 6 AS 1 I 0 found captive portal session
```

```
172.16.1.2-43490 > 192.168.0.10-8081 6 AS 1 I 0 returning captive portal session
```

```
172.16.1.2-43490 > 192.168.0.10-8081 6 AS 1 I 0 found active binding for user_id 1
```

```
172.16.1.2-43490 > 192.168.0.10-8081 6 AS 1 I 0 matched auth rule id = 1 user_id = 1 realm_id = 2
```

# The user\_map\_query script

```
root@FTD# user_map_query.pl -h
```

## Options:

```
--dump-data <pre_str>    Dumps all troubleshooting data for user/group mapping. If provided,
                           the output files will be prepended with "<pre_str>_"
-d, --debug              enable debug logging (off by default)
-g, --group              Displays the users associated to the group(s) specified (can not be
                           passed with -i or -u)
-h, -?, --help          Print usage information
-i, --ip-addr           Displays the users associated to the IPv4 address(es) specified (can
                           not be passed with -g or -u)
--iu                    Include unified file data
--outfile               Dumps the output to the specified file
-s, --snort             Include data from snort's mapping
-u, --user              Displays the IP addresses associated to the user(s) specified (can
                           not be passed with -g or -i)
--unified-all          Displays all of the unified data per record regardless of the type
                           of query
--unified-dir           The directory to look for unified files (default is
                           /var/sf/user_enforcement)
--use-id                Treats the values passed as IDs (only relevant for user and group
                           queries)
```

Collect All Data

Troubleshoot Live

# Finding who that User ID belongs to

```
root@FTD# user_map_query.pl --use-id -u 1
```

```
Current Time: 01/17/2019 15:54:38 UTC
```

```
Getting information on username(s)...
```

```
User #1: test1 ← Username
```

```
ID:          1
Last Seen:   Unknown
for_policy:  0
Realm ID:    2
```

```
=====
|           Database           |
=====
```

```
##) IP Address [Realm ID] ← Currently Mapped IP Address(s)
1) ::ffff:172.16.1.2 [2]
```

```
##) Group Name (ID) ← Groups user belongs to
1) Test (3)
```

# Comparing Database and Snort output

```
root@FTD/home/admin# user_map_query.pl -s -u test1
```

```
Would you like to dump user data from snort now? (Current Time: 01/17/2019 16:08:03 UTC) [y,n]: y
Successfully commanded snort.
Current Time: 01/17/2019 16:08:05 UTC
Getting information on username(s)...
```

```
-----
User #1: test1
```

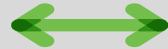
```
----
```

```
ID:          1
Last Seen:   Unknown
for_policy:  0
Realm ID:    2
```

```
=====
|           Database           |
=====
```

```
=====
|           Snort              |
=====
```

```
##) IP Address [Realm ID]
1) ::ffff:172.16.1.2 [2]
```



```
##) IP Address [Realm ID] (instances)
1) ::ffff:172.16.1.2 [2] (instance 1)
```

```
##) Group Name (ID)
1) Test (3)
```

```
##) Group Name (ID) (instances)
1) Test (3) (instance 1)
```

# Collect data to give to TAC

```
root@FTD# user_map_query.pl --dump-data CiscoLive
```

```
Would you like to dump user data from snort now? (Current Time: 01/17/2022 17:44:27 UTC) [y,n]: y
```

```
Successfully commanded snort.
```

```
Current Time: 01/17/2022 17:44:30 UTC
```

```
Getting database dumps...
```

```
Dumping table user_group_map...Done
```

```
Dumping table realm_info...Done
```

```
Dumping table user_identities...Done
```

```
Dumping table user_group...Done
```

```
Dumping table estreamer_bookmark...Done
```

```
Dumping table current_user_ip_map...Done
```

```
Dumping table user_ip_map...Done
```

```
Dumping table user_identities...Done
```

```
Done getting database dumps.
```

```
Added /var/sf/user_enforcement/* files.
```

```
Added snort data dumps
```

```
Compressing data...Done!
```

```
File: /var/tmp/CiscoLive_utd.a76e92ea-aaab-11e7-be62-c7b57db57e79.1647747070.tar.gz
```

```
Cleaning up...Done!
```

Give this to TAC







# Captive Portal packet captures

Lina Capture → Tun1 Capture → TEST → Stop Tun1 Cap → Stop Lina Cap → Copy Lina Cap

```
> capture ins_captport interface inside buffer 1000000 match tcp host 172.16.1.2 any
> expert

root@FTD1:# tcpdump -i tun1 -s 1518 -w /ngfw/var/common/captive_portal.pcap
HS_PACKET_BUFFER_SIZE is set to 4.tcpdump:
listening on tun1, link-type RAW (Raw IP), capture size 1518 bytes

[TEST AUTHENTICATION]

^C
99 packets captured
99 packets received by filter
0 packets dropped by kernel

root@FTD1:# exit

> capture ins_captport stop
> copy /noconfirm /pcap capture:ins_captport ins_captport.pcap
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
398 packets copied in 0.80 secs
```

**Lina Capture location:** /mnt/disk0/ins\_captport.pcap

**Tun1 Capture location:** /ngfw/var/common/captive\_portal.pcap

# The captures at an initial glance



ins\_captport.pcap

No.	Destination	Source	Protocol	Length	Info
261	172.16.1.1	172.16.1.2	TCP	66	52441 → 885 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 W
262	172.16.1.2	172.16.1.1	TCP	66	885 → 52441 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=
263	172.16.1.1	172.16.1.2	TCP	54	52441 → 885 [ACK] Seq=1 Ack=1 Win=65536 Len=0
264	172.16.1.1	172.16.1.2	TCP	233	52441 → 885 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=
265	172.16.1.2	172.16.1.1	TCP	54	885 → 52441 [ACK] Seq=1 Ack=180 Win=15744 Len=0
266	172.16.1.2	172.16.1.1	TCP	723	885 → 52441 [PSH, ACK] Seq=1 Ack=180 Win=15744 Le
267	172.16.1.1	172.16.1.2	TCP	268	52441 → 885 [PSH, ACK] Seq=180 Ack=670 Win=65024
268	172.16.1.2	172.16.1.1	TCP	336	885 → 52441 [PSH, ACK] Seq=670 Ack=394 Win=16768
269	172.16.1.1	172.16.1.2	TCP	571	52441 → 885 [PSH, ACK] Seq=394 Ack=952 Win=64512
270	172.16.1.2	172.16.1.1	TCP	54	885 → 52441 [ACK] Seq=952 Ack=911 Win=17920 Len=0
273	172.16.1.2	172.16.1.1	TCP	816	885 → 52441 [PSH, ACK] Seq=952 Ack=911 Win=17920

Before b1td NAT



After b1td NAT

No.	Destination	Source	Protocol	Length	Info
63	169.254.0.1	169.254.3.88	TCP	52	52441 → 885 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 W
64	169.254.3.88	169.254.0.1	TCP	52	885 → 52441 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=
65	169.254.0.1	169.254.3.88	TCP	40	52441 → 885 [ACK] Seq=1 Ack=1 Win=65536 Len=0
66	169.254.0.1	169.254.3.88	TCP	219	52441 → 885 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=
67	169.254.3.88	169.254.0.1	TCP	40	885 → 52441 [ACK] Seq=1 Ack=180 Win=15744 Len=0
68	169.254.3.88	169.254.0.1	TCP	709	885 → 52441 [PSH, ACK] Seq=1 Ack=180 Win=15744 L
69	169.254.0.1	169.254.3.88	TCP	254	52441 → 885 [PSH, ACK] Seq=180 Ack=670 Win=65024
70	169.254.3.88	169.254.0.1	TCP	322	885 → 52441 [PSH, ACK] Seq=670 Ack=394 Win=16768
71	169.254.0.1	169.254.3.88	TCP	557	52441 → 885 [PSH, ACK] Seq=394 Ack=952 Win=64512
72	169.254.3.88	169.254.0.1	TCP	40	885 → 52441 [ACK] Seq=952 Ack=911 Win=17920 Len=0
73	169.254.3.88	169.254.0.1	TCP	802	885 → 52441 [PSH, ACK] Seq=952 Ack=911 Win=17920

Same ports

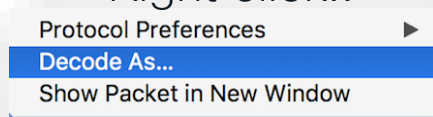


captive\_portal.pcap

# The captures may need to be decoded



Right click..



Choose SSL for each port

Field	Value	Type	Default	Current
TCP port	52441	Integer, base 10	(none)	SSL
TCP port	885	Integer, base 10	(none)	SSL

Raw

Protocol	Length	Info
TCP	52	52441 → 885 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SA
TCP	52	885 → 52441 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=14
TCP	40	52441 → 885 [ACK] Seq=1 Ack=1 Win=65536 Len=0
TCP	219	52441 → 885 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=179
TCP	40	885 → 52441 [ACK] Seq=1 Ack=180 Win=15744 Len=0
TCP	709	885 → 52441 [PSH, ACK] Seq=1 Ack=180 Win=15744 Len=669
TCP	254	52441 → 885 [PSH, ACK] Seq=180 Ack=670 Win=65024 Len=214
TCP	322	885 → 52441 [PSH, ACK] Seq=670 Ack=394 Win=16768 Len=282
TCP	557	52441 → 885 [PSH, ACK] Seq=394 Ack=952 Win=64512 Len=517
TCP	40	885 → 52441 [ACK] Seq=952 Ack=911 Win=17920 Len=0
TCP	802	885 → 52441 [PSH, ACK] Seq=952 Ack=911 Win=17920 Len=762



Decoded

Protocol	Length	Info
TCP	52	52441 → 885 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS
TCP	52	885 → 52441 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0
TCP	40	52441 → 885 [ACK] Seq=1 Ack=1 Win=65536 Len=0
TLSv1.2	219	Client Hello
TCP	40	885 → 52441 [ACK] Seq=1 Ack=180 Win=15744 Len=0
TLSv1.2	709	Server Hello, Certificate, Server Hello Done
TLSv1.2	254	Client Key Exchange, Change Cipher Spec, Finished
TLSv1.2	322	New Session Ticket, Change Cipher Spec, Finished
TLSv1.2	557	Application Data
TCP	40	885 → 52441 [ACK] Seq=952 Ack=911 Win=17920 Len=0
TLSv1.2	802	Application Data, Application Data

# Decrypting the captures provides even more insight



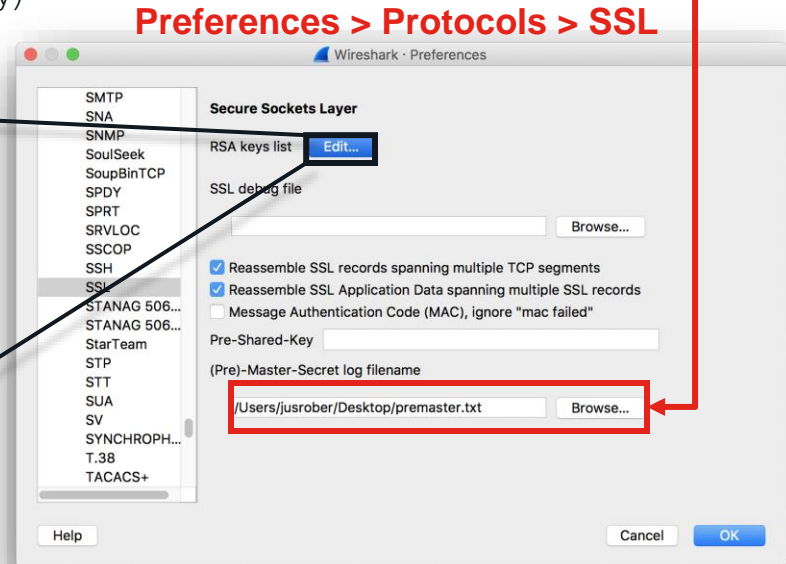
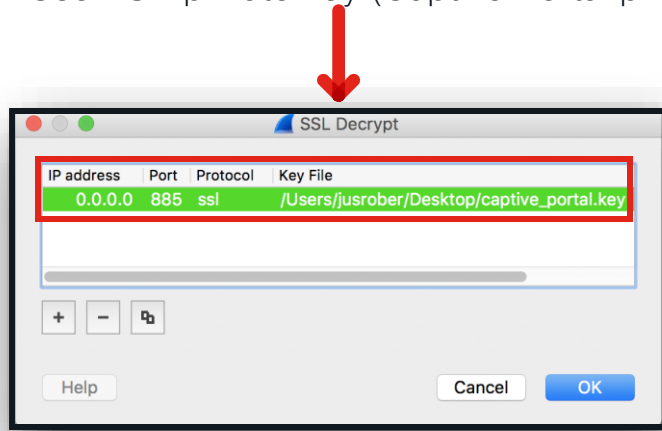
1. While testing captive portal, have sessions write out key information (Windows):

- Set environment variable to create a premaster secret file:

```
setx SSLKEYLOGFILE "%HOMEPATH%\Desktop\premaster.txt"
```

- Open a private / incognito window and test

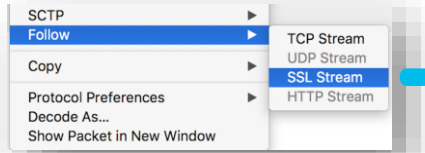
2. Use RSA private key (Captive Portal private key)



# You can now follow the SSL Stream



(Right click any SSL Packet)



Wireshark - Follow SSL Stream (tcp.stream eq 6) · a76e92ea-aaab-11e7-be62-c7b57db57e79-captive\_portal

```
GET /x.auth?s=gC7BnpEx3paFzafAeeoPYvGqq%2BI86qJ1cA4Piz6N4U%3D&u=http%3A%2F%2Fwww.cisco.com%2F HTTP/1.1
Host: 172.16.1.1:885
Connection: keep-alive
User-Agent: Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/63.0.3239.84 Safari/537.36
Upgrade-Insecure-Requests: 1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9

HTTP/1.1 401 Unauthorized
Date: Sat, 06 Jan 2018 20:53:12 GMT
Server: Apache
WWW-Authenticate: Basic realm="Please provide valid credentials"
Content-Length: 381
Keep-Alive: timeout=10, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>401 Unauthorized</title>
</head><body>
<h1>Unauthorized</h1>
<p>This server could not verify that you
are authorized to access the document
requested. Either you supplied the wrong
credentials (e.g., bad password), or your
browser doesn't understand how to supply
the credentials required.</p>
</body></html>

GET /x.auth?s=gC7BnpEx3paFzafAeeoPYvGqq%2BI86qJ1cA4Piz6N4U%3D&u=http%3A%2F%2Fwww.cisco.com%2F HTTP/1.1
Host: 172.16.1.1:885
Connection: keep-alive
Authorization: Basic VGVzdDE6UzB1cmZjFyMyE=
User-Agent: Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/63.0.3239.84 Safari/537.36
Upgrade-Insecure-Requests: 1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
```

2 client pkts, 1 server pkt, 2 turns.

Entire conversation (1623 bytes) Show and save data as ASCII

Find: Find Next

Help Filter Out This Stream Print Save as... Back Close

GET request  
after initial redirect

401 Unauthorized  
Challenge Response

Captured  
Credentials

# Redirect back to original destination



```
GET /x.auth?s=gC7BnpEx3paFZazfAeeoPYvGq%2BI86qJ1cA4Piz6N4U%3D&u=http%3A%2F%2Fwww.cisco.com%2F HTTP/1.1
Host: 172.16.1.1:885
Connection: keep-alive
Authorization: Basic VGVzdDE6UzB1cmMzZjFyMyE=
User-Agent: Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/63.0.3239.84 Safari/537.36
Upgrade-Insecure-Requests: 1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9

HTTP/1.1 307 Temporary Redirect
Date: Sat, 06 Jan 2018 20:53:22 GMT
Server: Apache
Location: http://www.cisco.com/
Content-Length: 231
Keep-Alive: timeout=10, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=iso-8859-1

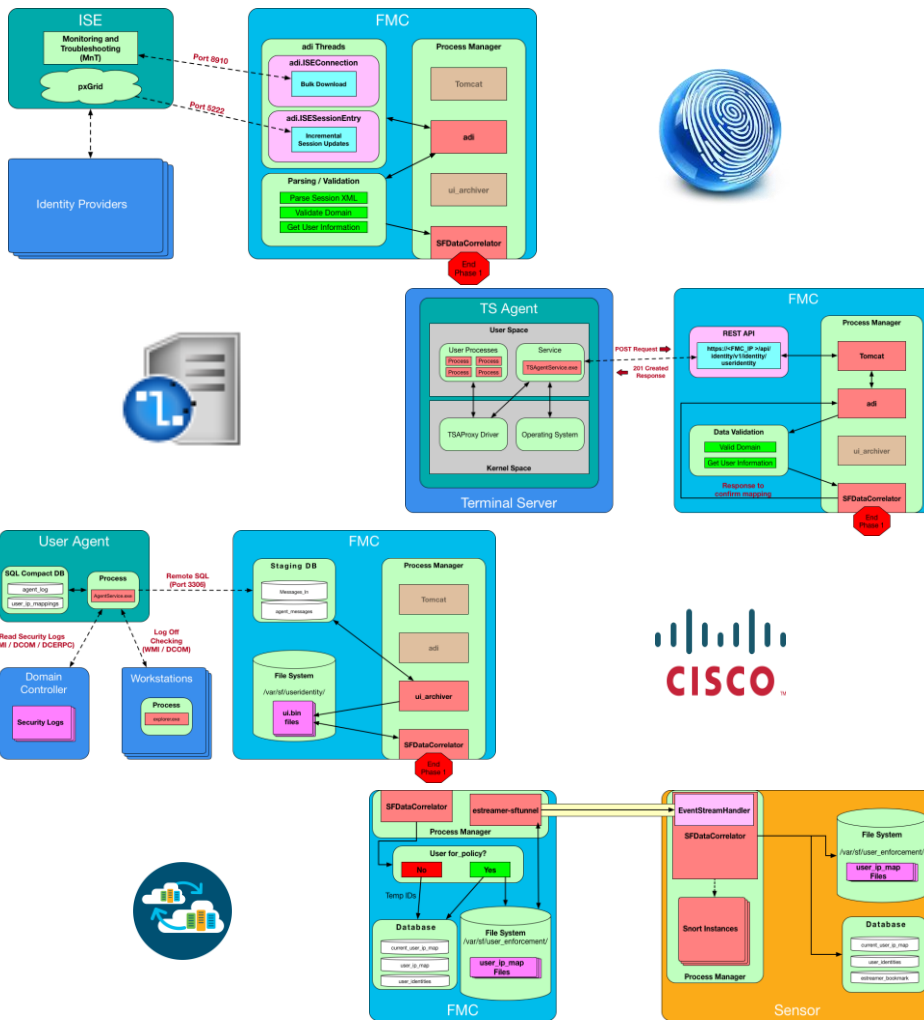
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>307 Temporary Redirect</title>
</head><body>
<h1>Temporary Redirect</h1>
<p>The document has moved <a href="http://www.cisco.com/">here</a>.</p>
</body></html>
```

← Original Destination

# Want more on Identity?

BRKSEC-3227

## Integrating & Troubleshooting Identity Features on the Firepower System



# Gathering Data For TAC





# Cisco Support Diagnostics (CSD)

## Smart License Status

Cisco Smart Software Manager



Usage Authorization:	Authorized (Last Synchronized On May 18 2019)
Product Registration:	Registered (Last Renewed On May 18 2019)
Assigned Virtual Account:	FTD-ENG-BLR
Export-Controlled Features:	Enabled
Cisco Success Network:	<u>Enabled</u>
Cisco Support Diagnostics:	<u>Enabled</u>

*No more TAC requests for  
troubleshoots!!*

## Cisco Cloud Services



The Management Center establishes a secure connection to the Cisco Cloud so that it can participate in additional service offerings from Cisco. Management Center will establish and maintain this secure connection at all times. You can turn off this connection at any time by disabling Cisco Success Network and Cisco Proactive Support. Disabling these services will disconnect the device from the cloud.

The Cisco Success Network provides usage information and statistics to Cisco. This information allows Cisco to improve the product and to make you aware of unused available features so that you can maximize the value of the product in your network. Check out the [sample data](#) that will be sent to Cisco.

Enable Cisco Success Network

The Cisco Proactive Support capability provides entitled customers with an enhanced support experience by allowing Cisco TAC to collect essential information from your devices during the course of a TAC case. Additionally, Cisco will periodically collect configuration and operational health data from your devices and process that data through our automated problem detection system, and proactively notify you of issues detected.

For more information, please review the data collection sheet located [here](#).

Enable Cisco Proactive Support

OK Cancel

# The First Responder Script



## SR 699999999 Cisco First Responder

To help us assist you with SR 699999999, in the most efficient manner possible, please run the following commands on your FMC and/or FTD devices in question.

1. Connect to the device using SSH
2. Issue the command `expert`, skip this step for FMC version 6.4.x and earlier
3. Issue the command `sudo su`
4. When prompted for the password, enter your password.
5. For version 6.4 and later issue the command

```
curl https://cxd.cisco.com/public/ctfr/firepower.py | python --c 699999999 -t LKJHdjkhalsdlkj --auto-upload &
```

6. For version 6.3.x and earlier issue the command

```
curl -k https://cxd.cisco.com/public/ctfr/firepower.py | python --c 699999999 -t LKJHdjkhalsdlkj --auto-upload &
```

Following the above steps will perform the below tasks silently:

1. Connect to Cisco's Customer eXperience Drive (CXD) and download a python script.
2. The python script downloaded will be run to
  - a. Collect a troubleshooting file and upload it to the case.
  - b. Search for any core files generated within the past 30 days, and upload them to the case.

If you would like to be prompted to select core files to upload please replace the command in the last step with `curl https://cxd.cisco.com/public/ctfr/firepower.py | python --c 699999999 -t LKJHdjkhalsdlkj`

The `-k` switch used with the `curl` command means `curl` will not verify the signing certificate, this is needed for any FMC/FTD version prior to 6.4 since the root certificate used by CXD was not trusted by Firepower devices until version 6.4.

For 6.3 and earlier versions we recommend confirming `cxd.cisco.com` resolves to 72.163.14.108 or 173.37.151.76. Furthermore, we recommend validating the SHA checksum of the file by running `curl -s -k https://cxd.cisco.com/public/ctfr/firepower.py | shasum` which should output 973058564a549d9a2dc6cc7e5dbdb42954ae2093.

If you would like to inspect the script code, [click here](#) to download the script.

Troubleshoot files are a common key first step in understanding your setup and subsequently assisting our valued customers troubleshooting and resolving their issues, we thank you for your cooperation.

This is an automatically generated message.

\*\*\* Troubleshoot File \*\*\*

- \* Connect to the device using SSH
- \* Issue the command `expert`, skip this step for FMC version 6.4.x and earlier
- \* Issue the command `sudo su`
- \* When prompted for the password, enter your password.
- \* For version 6.4 and later issue the command

```
curl https://cxd.cisco.com/public/ctfr/firepower.py | python --c 699999999 -t LKJHdjkhalsdlkj --auto-upload &
```

- \* For version 6.3.x and earlier issue the command

```
curl -k https://cxd.cisco.com/public/ctfr/firepower.py | python --c 699999999 -t LKJHdjkhalsdlkj --auto-upload &
```

Following the above steps will perform the below tasks silently:

- \* Connect to Cisco's Customer eXperience Drive (CXD) and download a python script.
- \* The python script downloaded will be run to Collect a troubleshooting file and upload it to the case.
- \* Search for any core files generated within the past 30 days, and upload them to the case.

If you would like to be prompted to select core files to upload please replace the command in the last step with `curl https://cxd.cisco.com/public/ctfr/firepower.py | python --c 699999999 -t LKJHdjkhalsdlkj`

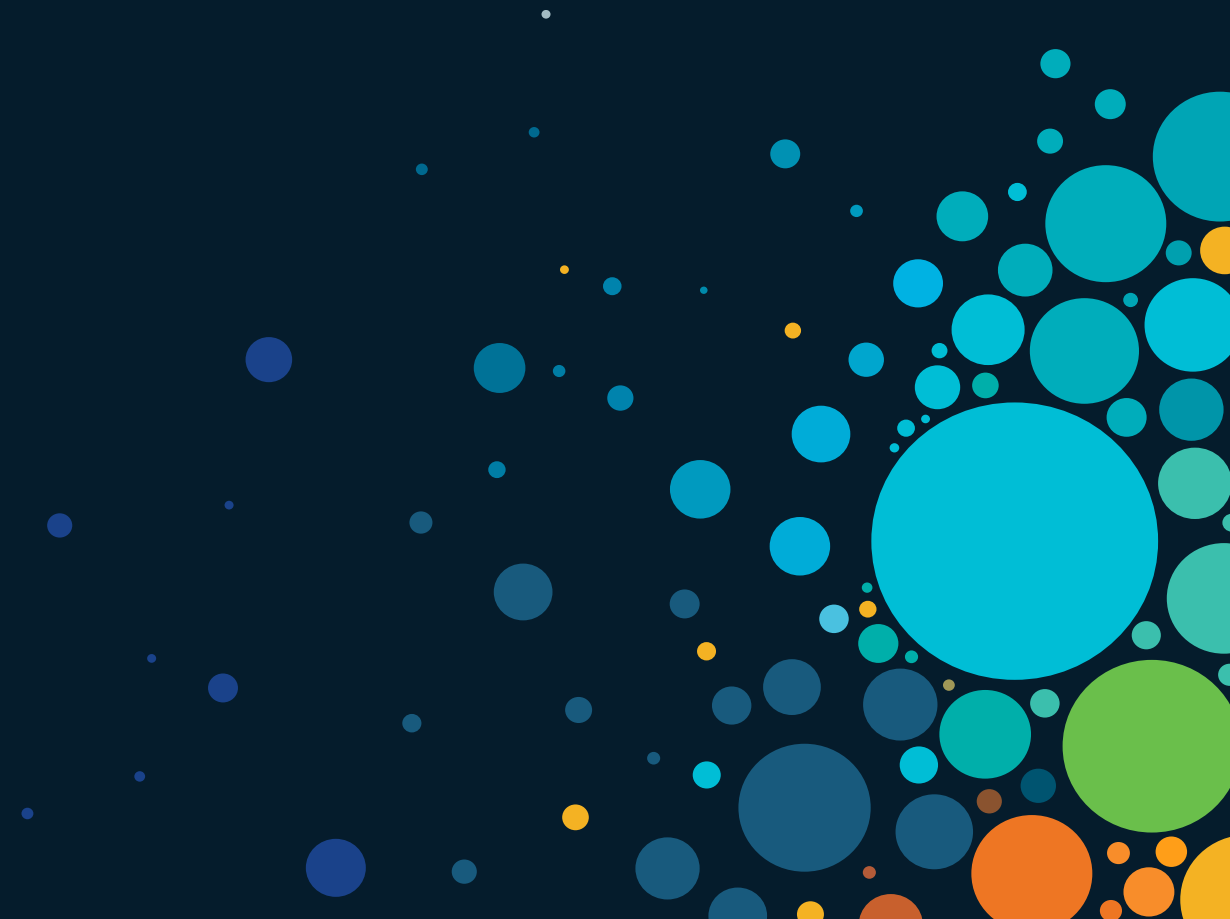
The `-k` switch used with the `curl` command means `curl` will not verify the signing certificate, this is needed for any FMC/FTD version prior to 6.4 since the root certificate used by CXD was not trusted by Firepower devices until version 6.4.

For 6.3 and earlier versions we recommend confirming `cxd.cisco.com` resolves to 72.163.14.108 or 173.37.151.76. Furthermore, we recommend validating the SHA checksum of the file by running `curl -s -k https://cxd.cisco.com/public/ctfr/firepower.py | shasum` which should output 973058564a549d9a2dc6cc7e5dbdb42954ae2093.

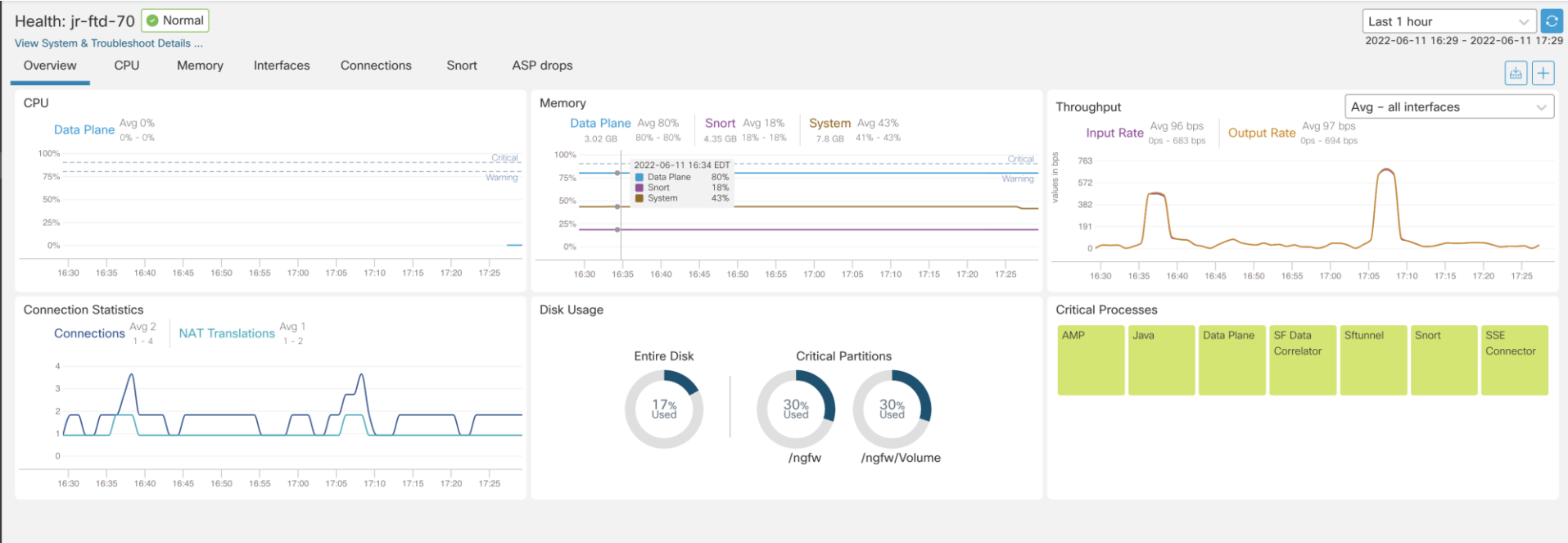
If you are unable to upload troubleshooting files (or would prefer not to), please let us know what hardware and software version you are running if you have not already .If you would like to inspect the script code, you can view it here: <https://cxd.cisco.com/public/ctfr/firepower.py>.

Sincerely, First Responder Team

New tools



# Device Health Monitoring



# Elephant Flow Detection



## Elephant Flow Settings

**i** For Snort 3 FTD devices 7.2.0 onwards, use this window to configure elephant flow.  
For all Snort 2 FTD devices or Snort 3 FTD devices 7.1.0 and earlier, use the Intelligent Application Bypass settings.

Elephant flow detection does not apply to encrypted traffic. [Learn more](#)

### Elephant Flow Detection



Generate elephant flow events when flow bytes **exceeds**  MB and flow duration **exceeds**  seconds

<input type="checkbox"/>	↓ First Packet	Last Packet	Action	Reason	Initiator IP	Initiator Country	Responder IP	Responder Country	Ingress Security Zone	Egress Security Zone	Source Port
▼ <input type="checkbox"/>	2022-03-11 17:13:05	2022-03-11 17:14:46	Allow	Elephant Flow	<input type="text" value="10.69.2.3"/>		<input type="text" value="10.69.1.5"/>		inside2	inside1	45988 / tcp
▼ <input type="checkbox"/>	2022-03-11 17:13:05		Allow	Elephant Flow	<input type="text" value="10.69.2.3"/>		<input type="text" value="10.69.1.5"/>		inside2	inside1	45988 / tcp

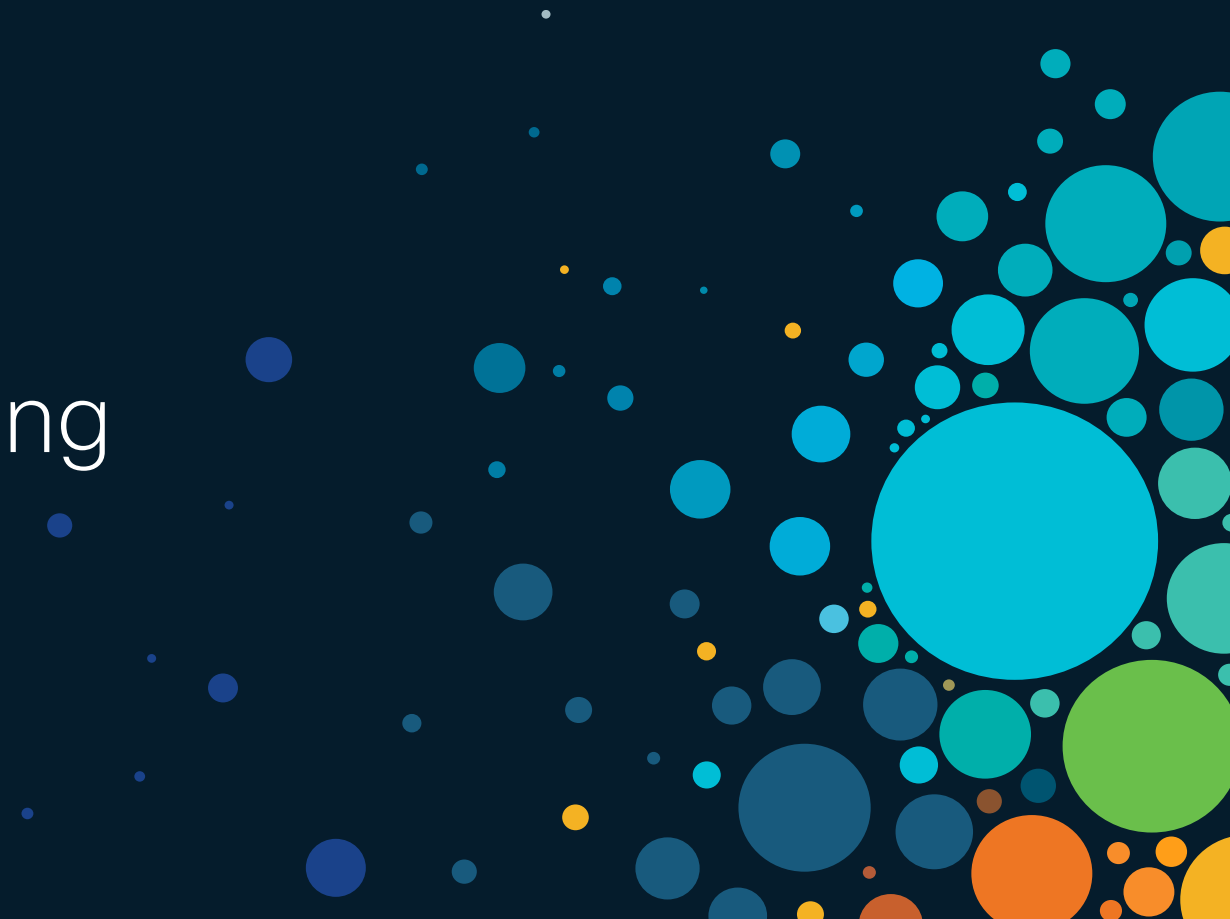
# Unified Event Viewer

[Refresh](#)

Showing all 35 events (🔍 33 📄 1 🚩 1) 🕒 Last 1 hour [Go Live](#)

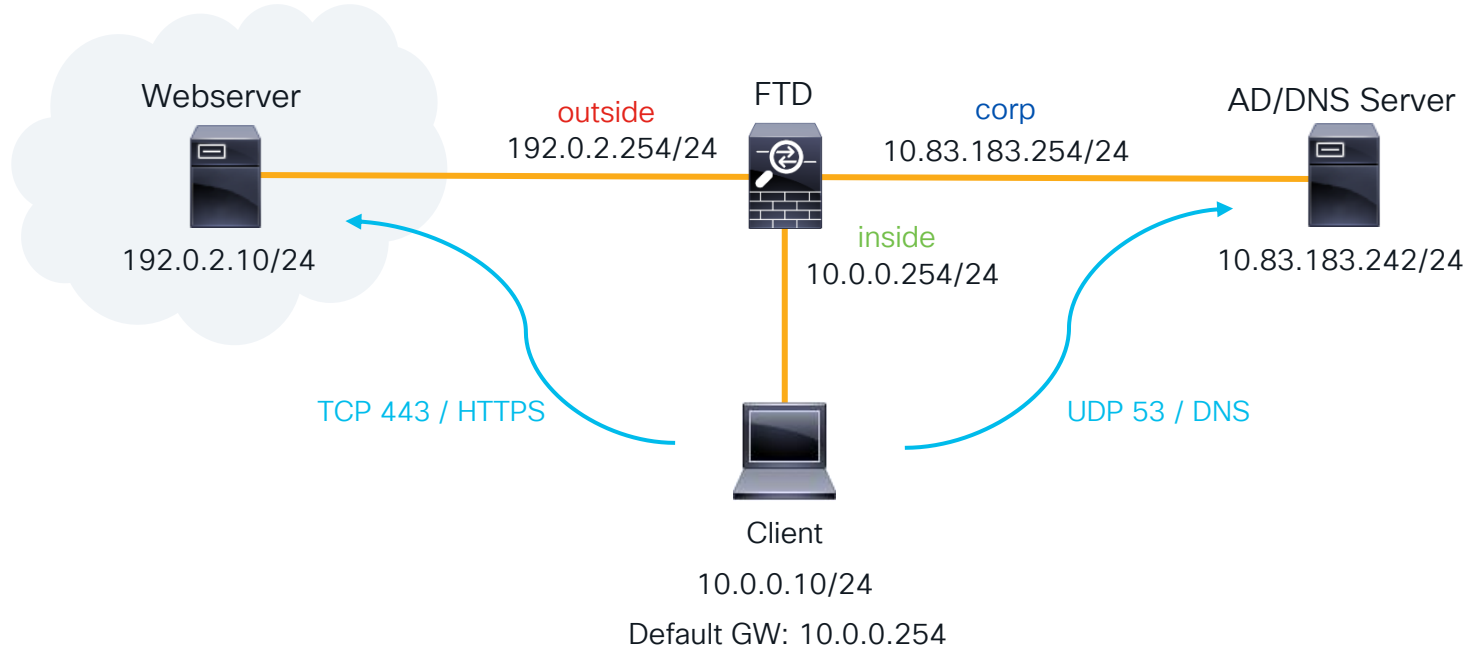
Time	Event Type	Action	Reason	Source IP	Destination IP	Source Port / ICMP Type	Destination Port / ICMP Code	Web Application	Access Control Rule	Access Control Policy	Dev
2022-06-11 17:34:41	🔗 Connection	🛑 Block	File Block	192.168.70.3	10.83.180.17	58504 / tcp	80 (http) / tcp	Web Browsing	Inspection	lab_policy	jr-f
2022-06-11 17:34:41	📄 File	Malware Block		10.83.180.17	192.168.70.3	80 (http) / tcp	58504 / tcp	Web Browsing			jr-f
2022-06-11 17:34:41	🚩 Malware	Malware Block		10.83.180.17	192.168.70.3	80 (http) / tcp	58504 / tcp	Web Browsing			jr-f
2022-06-11 17:34:33	🔗 Connection	✅ Allow		192.168.70.3	10.83.180.17	58502 / tcp	80 (http) / tcp	Web Browsing	Inspection	lab_policy	jr-f
2022-06-11 17:34:06	🔗 Connection	✅ Allow		192.168.70.3	10.83.180.17	58490 / tcp	80 (http) / tcp	Web Browsing	Inspection	lab_policy	jr-f
2022-06-11 17:34:06	🔗 Connection	✅ Allow		192.168.70.3	10.83.180.17	58488 / tcp	80 (http) / tcp	Web Browsing	Inspection	lab_policy	jr-f
2022-06-11 17:34:06	🔗 Connection	✅ Allow		192.168.70.3	10.83.180.17	58484 / tcp	80 (http) / tcp	Web Browsing	Inspection	lab_policy	jr-f
2022-06-11 17:34:06	🔗 Connection	✅ Allow		192.168.70.3	10.83.180.17	58482 / tcp	80 (http) / tcp	Web Browsing	Inspection	lab_policy	jr-f
2022-06-11 17:34:06	🔗 Connection	✅ Allow		192.168.70.3	10.83.180.17	58500 / tcp	80 (http) / tcp	Web Browsing	Inspection	lab_policy	jr-f
2022-06-11 17:34:06	🔗 Connection	✅ Allow		192.168.70.3	10.83.180.17	58498 / tcp	80 (http) / tcp	Web Browsing	Inspection	lab_policy	jr-f
2022-06-11 17:34:06	🔗 Connection	✅ Allow		192.168.70.3	10.83.180.17	58496 / tcp	80 (http) / tcp	Web Browsing	Inspection	lab_policy	jr-f
2022-06-11 17:34:06	🔗 Connection	✅ Allow		192.168.70.3	10.83.180.17	58494 / tcp	80 (http) / tcp	Web Browsing	Inspection	lab_policy	jr-f
2022-06-11 17:34:06	🔗 Connection	✅ Allow		192.168.70.3	10.83.180.17	58492 / tcp	80 (http) / tcp	Web Browsing	Inspection	lab_policy	jr-f
2022-06-11 17:34:06	🔗 Connection	✅ Allow		192.168.70.3	10.83.180.17	58486 / tcp	80 (http) / tcp	Web Browsing	Inspection	lab_policy	jr-f
2022-06-11 17:34:06	🔗 Connection	✅ Allow		192.168.70.3	10.83.180.17	58480 / tcp	80 (http) / tcp	Web Browsing	Inspection	lab_policy	jr-f
2022-06-11 17:34:06	🔗 Connection	✅ Allow		192.168.70.3	10.83.180.17	58478 / tcp	80 (http) / tcp	Web Browsing	Inspection	lab_policy	jr-f
2022-06-11 17:33:46	🔗 Connection	🛑 Block	Intrusion Block	192.168.70.3	10.83.180.17	58476 / tcp	80 (http) / tcp		Inspection	lab_policy	jr-f
2022-06-11 17:33:12	🔗 Connection	✅ Allow		192.168.70.3	10.83.180.17	58472 / tcp	80 (http) / tcp	Web Browsing	Inspection	lab_policy	jr-f
2022-06-11 17:33:12	🔗 Connection	✅ Allow		192.168.70.3	10.83.180.17	58470 / tcp	80 (http) / tcp	Web Browsing	Inspection	lab_policy	jr-f

# Interactive Troubleshooting



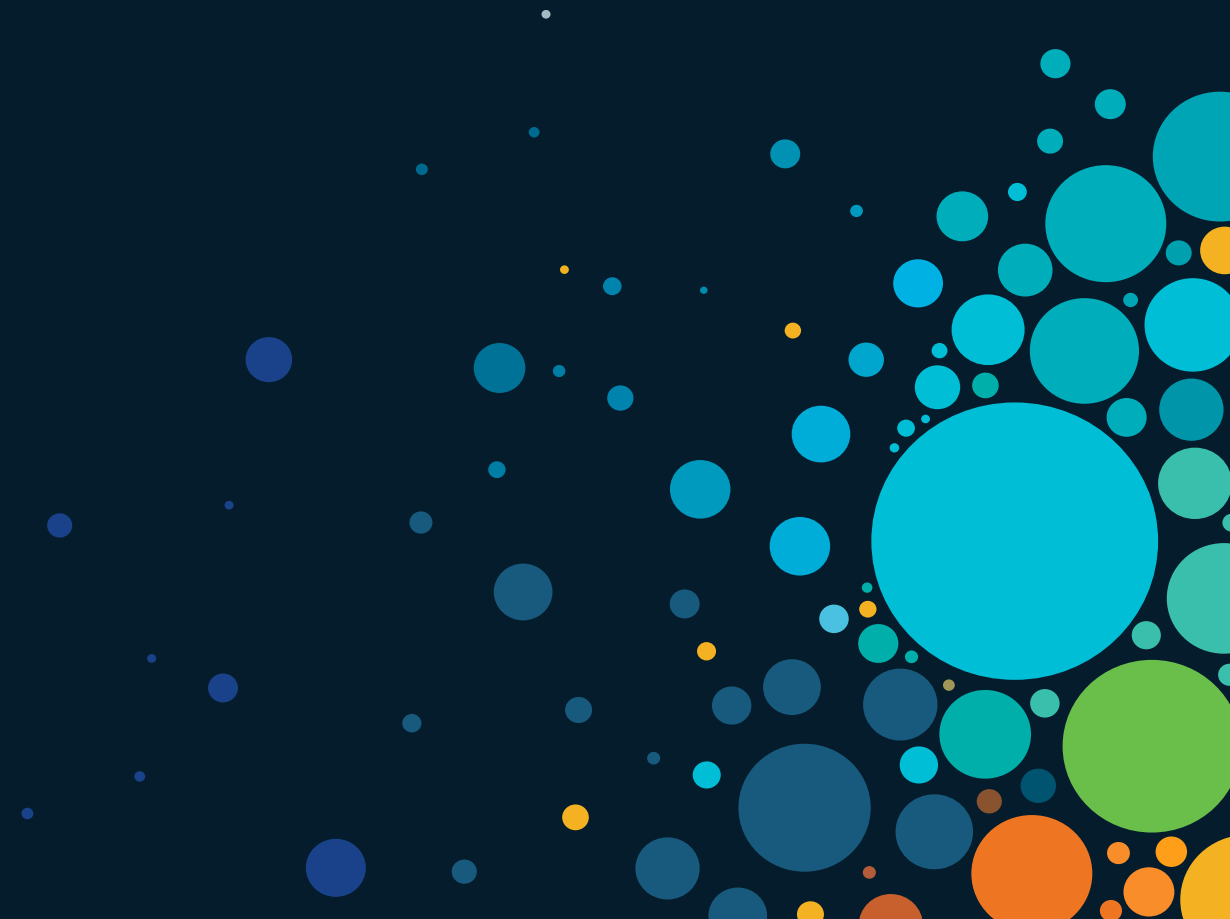
# Scenario Topology

- Goal: Client to retrieve a file from an external webserver via HTTPS through FTD





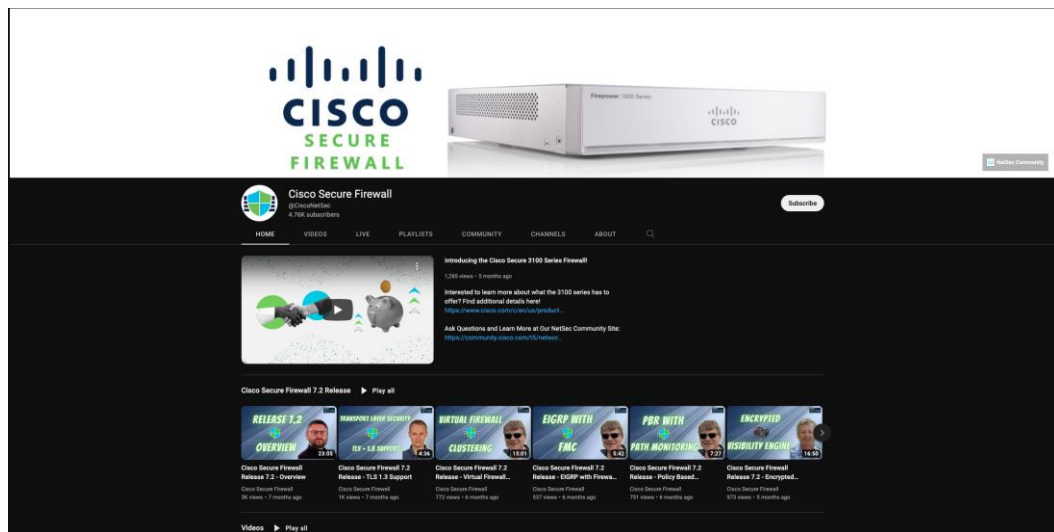
# Questions & Answers



# Cisco Secure Firewall Youtube

Knowledge from TAC / TMEs

- New Feature Walkthroughs
- Troubleshooting Tips
- Automation Guides



# Security Beta Programs



**Beta Software  
Access**



**Product  
Training**



**Access to  
Dev Teams**



**Test Hardware  
and Licenses**



**Bugs Fixed for  
Release**



**Influence  
Product Roadmap**

Presented By **Cisco Security Customer Insights**

ASA | NGFW | NGIPS | Firepower Platforms | AMP | CTA | ESA | WSA | ISE | Umbrella  
Enroll today!

<http://cs.co/security-beta-nomination>  
[ask-sbg-beta@cisco.com](mailto:ask-sbg-beta@cisco.com)



*"I've been involved in many beta programs ...  
I must say that this one has been the best organized.  
This beta has taken a very active, hands-on  
approach."* - **Liberal Arts College Customer**

# Wrapping it up

- Apply new skills to your daily FTD troubleshooting.
- Check out the additional resources and slides for future reference purposes.
- Although FTD is complex, you should now have a better understanding of the product architecture, traffic flow, and troubleshooting tools that are available to help you quickly resolve issues.
- If you leverage those newfound skills and resources, before you know it you'll be troubleshooting FTD like a TAC engineer!

# Security Technologies

## Next Generation Firewall

Learn how Cisco Secure Firewall keeps businesses moving while keeping it secure. They offer deep visibility using built-in advanced security features like Cisco Secure IPS and Cisco Secure Endpoint to detect and stop advanced threats.



START

Feb 5 | 16:45  
**LABSEC-2030**  
Firepower Threat Defense: identity based firewall for VPN remote users - configuration and troubleshooting

Feb 5 | 16:45  
**LABSEC-2334**  
Deploying Cisco NGFW in Public Cloud (AWS).

Feb 5 | 18:15  
**LABSEC-1671**  
Adaptive Network Control with ISE and FTD

Feb 5 | 19:00  
**LABSEC-3449**  
Implementing and troubleshooting SAML authentication for AnyConnect VPN users terminated on Firepower Threat Defense

Feb 6 | 08:45  
**TECSEC-3782**  
Troubleshooting Firepower Threat Defense like a TAC Engineer

Feb 7 | 08:30  
**BRKSEC-1018**  
Introduction to cloud-delivered Firewall Management Center

Feb 7 | 3:30  
**BRKSEC-2109**  
Traffic Inspection in Azure Cloud Environment using Cisco Secure Firewall and Gateway Load Balancing

Feb 7 | 14:45  
**BRKSEC-1138**  
Security Management from Ar Cisco Defense Orchestrator & Analytics and Logging

Feb 8 | 08:30  
**BRKSEC-2236**  
Security Management on network Security with Cisco Secure Firewall

Feb 8 | 08:30  
**LTRSEC-3391**  
Secure Firewalls in ACI Deep Dive Lab

Feb 8 | 09:00  
**PSOSEC-1211**  
Cisco Secure Firewall: Driving Security Resilience Across a Hybrid and Multicloud World

Feb 8 | 13:30  
**BRKSEC-2484**  
Snort 3 with the Cisco Secure Firewall

Feb 8 | 16:45  
**BRKSEC-2201**  
SecureX and Secure Firewall Better Together

Feb 8 | 17:00  
**BRKSEC-2123**  
Solving the Segmentation Puzzle! Secure Workload and Secure Firewall Integration

Feb 9 | 08:30  
**BRKSEC-3320**  
Demystifying TLS Decryption and Encrypted Visibility Engine on Cisco Secure Firewall Threat Defense

Feb 9 | 14:00  
**LTRSEC-2735**  
Deploying Cisco Firewalls in the Azure Public Cloud

Feb 9 | 15:45  
**BRKSEC-3058**  
Route based VPNs with Cisco Secure Firewall

Feb 10 | 11:15  
**BRKSEC-3533**  
Think Like a TAC Engineer: A guide to Cisco Secure Firewall most common pain points

FINISH

**HIGHLY RECOMMENDED**

# Complete your Session Survey

- Please complete your session survey after each session. Your feedback is very important.
- Complete a minimum of 4 session surveys and the Overall Conference survey (open from Thursday) to receive your Cisco Live t-shirt.
- All surveys can be taken in the Cisco Events Mobile App or by logging in to the Session Catalog and clicking the "Attendee Dashboard" at <https://www.ciscolive.com/emea/learn/sessions/session-catalog.html>



# Continue Your Education



Visit the Cisco Showcase for related demos.



Book your one-on-one Meet the Engineer meeting.



Attend any of the related sessions at the DevNet, Capture the Flag, and Walk-in Labs zones.



Visit the On-Demand Library for more sessions at [ciscolive.com/on-demand](https://ciscolive.com/on-demand).



The bridge to possible

Thank you

CISCO *Live!*



CISCO *Live!*

ALL IN

# Appendix

# Troubleshooting Walkthroughs



# Scenario 1: Facebook is not blocked as expected and CNN is unexpectedly being blocked



The customer on 10.1.1.10 is able to access Facebook.com, whereas this client should be blocked from all Social Networking sites.

The customer's Access Control Policy is many pages long!

Let's troubleshoot this using a systematic approach to FTD troubleshooting

# Using our FTD troubleshooting tools

Remember: Always check events and syslogs! FMC: Analysis → Connections → Events

No connection events for 10.1.1.10 navigating to Facebook. We must not be logging the rule which allows it.

7	Block Auction URL	Any	Any	Any	Any	Any	Any	Any	Any	Any	Any	Auctions (Any Re	Any	Block
8	Block Games URL	Any	Any	Any	Any	Any	Any	Any	Any	Any	Any	Games (Any Repr	Any	Block
9	Block Hacking URL	Any	Any	Any	Any	Any	Any	Any	Any	Any	Any	Hacking (Any Re	Any	Block
10	Block Job Search URL	Any	Any	Any	Any	Any	Any	Any	Any	Any	Any	Job Search (Any	Any	Block
11	Block Malware URL	Any	Any	Any	Any	Any	Any	Any	Any	Any	Any	Malware Sites (A	Any	Block
12	Block Parked Domains URL	Any	Any	Any	Any	Any	Any	Any	Any	Any	Any	Parked Domains	Any	Block
13	Block Social Networking URL	Any	Any	Any	Any	Any	Any	Any	Any	Any	Any	Social Network (A	Any	Block

The rule we expect traffic to hit

# Firewall engine debug

At this point, we suspect there is a problem with rule evaluation.

Firewall Engine Debug is the right tool to identify what is happening within the Access Control Policy

```
> system support firewall-engine-debug
```

```
Please specify an IP protocol: tcp
```

```
Please specify a client IP address: 192.168.1.10
```

```
Please specify a client port:
```

```
Please specify a server IP address:
```

```
Please specify a server port: 443
```

```
Monitoring firewall engine debug messages
```

```
192.168.1.10-49986 > 31.13.69.228-443 6 AS 1 I 1 New session
```

```
192.168.1.10-49986 > 31.13.69.228-443 6 AS 1 I 1 Starting with minimum 2, 'Allow Facebook', and SrcZone first with zones 4 -> 3, geo 0(0) -> 0, vlan 0, inline sgt tag: untagged, ISE sgt id: 0, svc 1122, payload 629, client 1296, misc 0, user 9999997, url facebook.com, xff
```

```
192.168.1.10-49986 > 31.13.69.228-443 6 AS 1 I 1 match rule order 2, 'Allow Facebook', action Allow
```

```
192.168.1.10-49986 > 31.13.69.228-443 6 AS 1 I 1 allow action
```

Whoops... we must have forgotten about an earlier rule.

# Revisiting the Access Control Policy

Rule 2 (Allow application Facebook) is not logging, so connection events are not generated

1	Trust Backup Servers	Any	Any	📄 10.	📄 10.	Any	Any	Any	Any	Any	Any	Any	Any	Any	→ Trust	🛡️	📄
2	Allow Facebook	Any	Any	Any	Any	Any	Any	Facebook	Any	Any	Any	Any	Any	Any	✓ Allow	🛡️	📄
3	Block Example.com	Any	Any	Any	Any	Any	Any	example.com	Any	Any	Any	Any	Any	Any	✗ Block	🛡️	📄
4	Block Gambling Sites	Any	Any	Any	Any	Any	Any	Gambling (Any R	Any	Any	Any	Any	Any	Any	✗ Block	🛡️	📄
5	Safesearch test	Any	Any	Any	Any	Any	Any	Any	Any	Any	Any	Any	Any	Any	✓ Allow	🛡️	📄
6	File Inspection	Any	Any	Any	Any	Any	Any	Any	Any	Any	Any	Any	Any	Any	✓ Allow	🛡️	📄
7	Block Auction URL	Any	Any	Any	Any	Any	Any	Any Re	Any	Any	Any	Any	Any	Any	✗ Block	🛡️	📄
8	Block Games URL	Any	Any	Any	Any	Any	Any	Games (Any Repr	Any	Any	Any	Any	Any	Any	✗ Block	🛡️	📄

Key Takeaway: Firewall Engine Debug shows rule evaluation, even if logging is not enabled

# Check Application Categories and Tags

Connection  
Events



First Packet	Last Packet	Action	Initiator IP	Responder IP	Source Port / ICMP Type	Destination Port / ICMP Code	Application Protocol	Web Application	Application Risk	Business Relevance	URL
2017-05-19 16:02:29		Block	192.168.62.63	151.101.65.67	54308 / tcp	80 (http) / tcp	HTTP	CNN.com	Medium	Medium	http://cnn.com/

**CNN.com**

Turner Broadcasting System's news website.

<b>Type</b>	Web Application
<b>Risk</b>	Very Low
<b>Business Relevance</b>	High
<b>Categories</b>	multimedia (TV/video), news
<b>Tags</b>	displays ads

Context Explorer Wikipedia Google Yahoo! Bing



# Check Application Categories and Tags

firewall-  
engine-debug



```
192.168.62.63-54308 > 151.101.65.67-80 6 AS 1 I 0 New session
192.168.62.63-54308 > 151.101.65.67-80 6 AS 1 I 0 Starting with minimum 4, 'block by category', and SrcZone
first with zones 1 -> 2, geo 0 -> 0, vlan 0, inline sgt tag: untagged, ISE sgt id: 0, svc 0, payload 0,
client 0, misc 0, user 9999997, icmpType 0, icmpCode 0
192.168.62.63-54308 > 151.101.65.67-80 6 AS 1 I 0 pending rule order 4, 'block by category', AppID
192.168.62.63-54308 > 151.101.65.67-80 6 AS 1 I 0 Starting with minimum 4, 'block by category', and SrcZone
first with zones 1 -> 2, geo 0 -> 0, vlan 0, inline sgt tag: untagged, ISE sgt id: 0, svc 0, payload 0,
client 0, misc 0, user 9999997, icmpType 0, icmpCode 0
192.168.62.63-54308 > 151.101.65.67-80 6 AS 1 I 0 pending rule order 4, 'block by category', AppID
192.168.62.63-54308 > 151.101.65.67-80 6 AS 1 I 0 Starting with minimum 4, 'block by category', and SrcZone
first with zones 1 -> 2, geo 0 -> 0, vlan 0, inline sgt tag: untagged, ISE sgt id: 0, svc 0, payload 0,
client 0, misc 0, user 9999997, icmpType 0, icmpCode 0
192.168.62.63-54308 > 151.101.65.67-80 6 AS 1 I 0 pending rule order 4, 'block by category', AppID
192.168.62.63-54308 > 151.101.65.67-80 6 AS 1 I 0 URL SI: ShmDBLookupURL("http://cnn.com/") returned 0
192.168.62.63-54308 > 151.101.65.67-80 6 AS 1 I 0 Starting with minimum 4, 'block by category', and SrcZone
first with zones 1 -> 2, geo 0(0) -> 0, vlan 0, inline sgt tag: untagged, ISE sgt id: 0, svc 676, payload
1190, client 638, misc 0, user 9999997, url http://cnn.com/, xff
192.168.62.63-54308 > 151.101.65.67-80 6 AS 1 I 0 match rule order 4, 'block by category', action Block
192.168.62.63-54308 > 151.101.65.67-80 6 AS 1 I 0 sending block response of 605 bytes
192.168.62.63-54308 > 151.101.65.67-80 6 AS 1 I 0 Deleting session
```

# Scenario 2: Network traffic failure through FTD

The customer states that FTD is causing network performance problems after a weekend migration from another vendor firewall

What we know:

1. The problem began right around the time users started arriving to the office.
2. Users are unable to open web sites.
3. The engineer is unable to join a WebEx.
4. The engineer states that Snort is “stuck at 100% utilization”

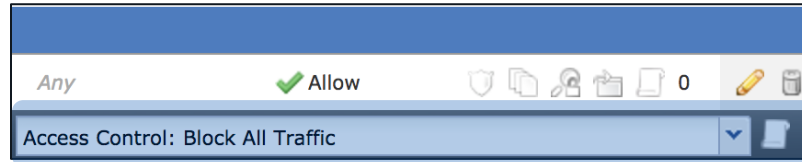
So, what does a “systemic approach to FTD troubleshooting” look like in this scenario?

# Network traffic failure through FTD

Step 1: Given the impact and since we have no access to troubleshoot directly, we enable a Prefilter policy for all traffic to temporarily stop sending traffic to Snort.

This alleviates the problem and the engineer is able to join a WebEx. Since a Prefilter policy improved the situation, we suspect a Snort oversubscription or policy issue.

Step 2: Visually review policy to determine what rule traffic would match



Customer had a “Block All” rule that was unintentional

What troubleshooting tool would have shown this without a visual inspection?

# Network traffic failure through FTD

Minutes later, intermittent connectivity issues continue. Engineer's PC loses connectivity to Exchange.

```
capture capin type raw-data buffer 33000000 trace interface Inside  
[Capturing - 25500768 bytes] match tcp host 10.0.10.1 any eq https
```

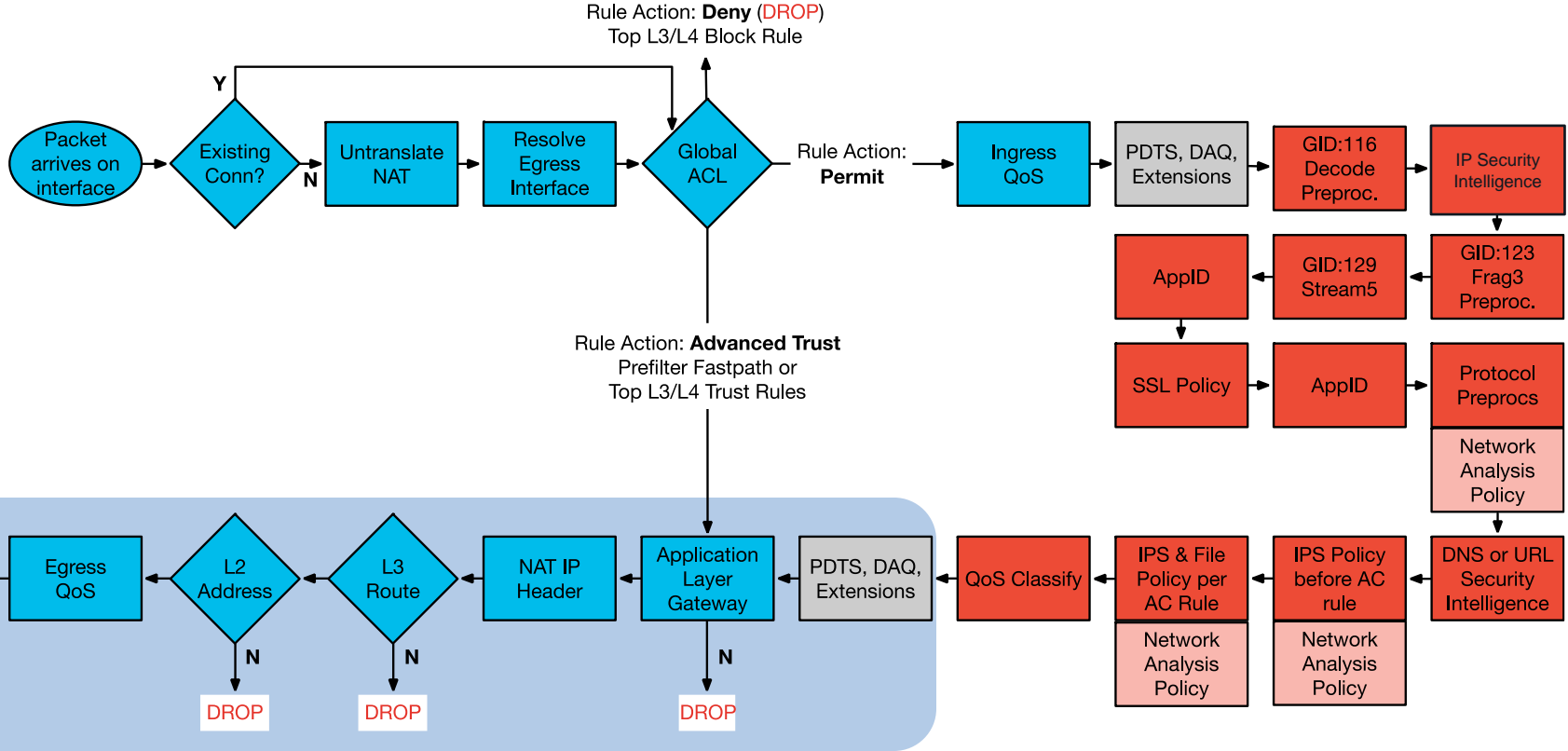
Enable capture for failing flow

```
firepower# sh cap capin | i S  
3: 13:23:11.905669 10.0.10.1.5377 > 192.0.2.194.443: S 2773524504:2773524504(0) win 8192  
19: 13:23:12.514499 10.0.10.1.5386 > 192.0.2.18.443: S 1117279318:1117279318(0) win 8192  
30: 13:23:12.797398 10.0.10.1.5379 > 192.0.2.98.443: S 3103152246:3103152246(0) win 8192  
32: 13:23:13.123650 10.0.10.1.5389 > 192.0.2.194.443: S 3496291677:3496291677(0) win 8192  
34: 13:23:13.163733 10.0.10.1.5387 > 192.0.2.194.443: S 3669311460:3669311460(0) win 8192  
43: 13:23:13.306411 10.0.10.1.5381 > 192.0.2.194.443: S 1115384746:1115384746(0) win 8192  
44: 13:23:13.446372 10.0.10.1.5390 > 192.0.2.194.443: S 3466698234:3466698234(0) win 8192
```

Identify instance of TCP connection attempt (SYN)

Based on what we learned today, what should we check next?

# Reference Slide: Routed FTD Path of Packet



LINA ASA Engine = BLUE

Snort Engine = RED

# Network traffic failure through FTD

Packet tracer output for affected traffic:

```
firepower# show capture capin trace pack 19
56752 packets captured

  19: 13:23:12.514499      10.0.10.1.5386 > 192.0.2.18.443: S 1117279318:1117279318(0) win
8192 Phase: 1
Type: CAPTURE
Subtype:
Result: ALLOW
Config:
Additional Information:
MAC Access list
...
Result:
input-interface: Inside
input-status: up
input-line-status: up
output-interface: Outside
output-status: up
output-line-status: up
Action: drop
Drop-reason: (nat-xlate-failed) NAT failed
```

Here we see that we have a NAT problem that is unrelated to Snort policy.

# Network traffic failure through FTD

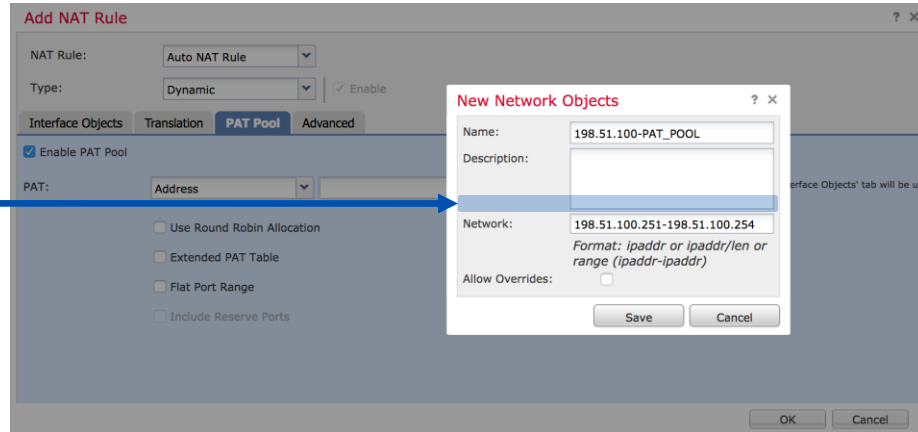
Check NAT pool allocations:

```
firepower# show nat pool
TCP PAT pool Outside, address 198.51.100.251, range 1-511, allocated 0
TCP PAT pool Outside, address 198.51.100.251, range 512-1023, allocated 0
TCP PAT pool Outside, address 198.51.100.251, range 1024-65535, allocated 64512
UDP PAT pool Outside, address 198.51.100.251, range 1-511, allocated 2
UDP PAT pool Outside, address 198.51.100.251, range 512-1023, allocated 0
UDP PAT pool Outside, address 198.51.100.251, range 1024-65535, allocated 23
firepower#
```

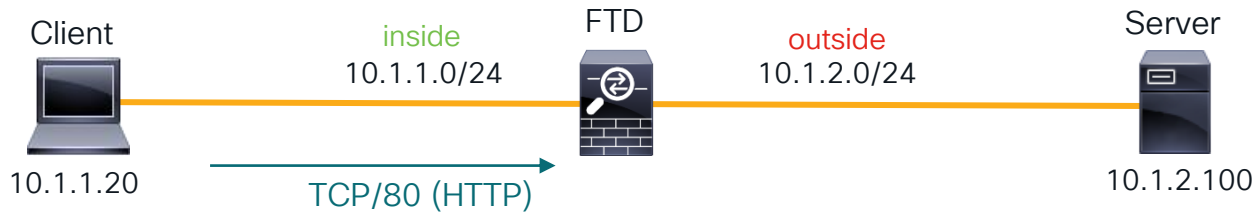
PAT pool port exhaustion

Solution:

Add more IP addresses to PAT pool



# Scenario 3: Clients cannot reach server



The customer states that clients traversing FTD are **not** able to access an internal web server. However, other clients on the server subnet (10.1.2.0/24) **are** able to access the server.

Let's troubleshoot this using a systematic approach to FTD troubleshooting



# Using our FTD troubleshooting tools

Remember: Always check events and syslogs! FMC: Analysis → Connections → Events

No events found! (Always make sure you're logging the rule that you expect to be hitting!)

Fortunately, we did enable Lina syslogs to an external server. Here's what we found:

```
%ASA-6-302013: Built inbound TCP connection 46927 for inside:10.1.1.20/2286 (10.1.1.20/2286) to outside:10.1.2.100/80 (10.1.2.100/80)
%ASA-6-302014: Teardown TCP connection 46927 for inside:10.1.1.20/2286 to outside:10.1.2.100/80 duration 0:00:30 bytes 0 SYN Timeout
```

So, now we know that we are receiving the packet but either the server is not responding or FTD is not forwarding it. Let's dig deeper. Maybe snort is dropping it...

```
> system support firewall-engine-debug
Please specify an IP protocol: tcp
Please specify a client IP address: 10.1.1.20
....
10.1.1.20-2286 > 10.1.2.100-80 6 AS 1 | 16 New session
10.1.1.20-2286 > 10.1.2.100-80 6 AS 1 | 16 using HW or preset rule order 5,
'Allow_Inside_to_Outside', action Allow and prefilter rule 0
10.1.1.20-2286 > 10.1.2.100-80 6 AS 1 | 16 allow action
```

It looks like  
Snort allows it.  
So what next?

# Packet Captures – The single source of truth

What do we know at this point?

FTD **is** receiving the packet. We **are** building the TCP connection for the flow. Snort is **NOT** dropping the packet.

The next step here is to determine if FTD is actually **forwarding** the packet.

Let's use our awesome packet capture tools for this.

Verify ingress captures so we can line them up with egress captures:

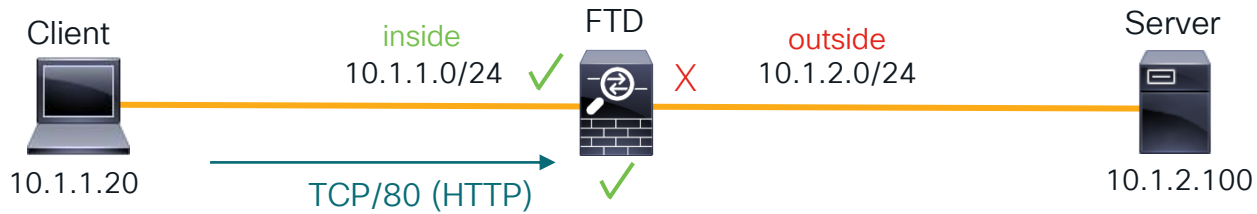
```
firepower# show capture
capture capin type raw-data trace interface inside [Buffer Full - 524216 bytes]
  match tcp host 10.1.1.20 host 10.1.2.100 eq www
firepower# sho cap capin | i 2286
322: 13:04:56.926786      802.1Q vlan#36 P0 10.1.1.20.2286 > 10.1.2.100.80: S
1336706021:1336706021(0) win 512
firepower#
```

```
capture capout type raw-data interface outside
[Capturing - 0 bytes]
match tcp any host 10.1.2.100 eq www
```

Houston...we have a problem.

No packets going to the destination server?

# Visual troubleshooting recap



- Packet **is** received
- Lina **is** building connection
- Snort is **not** dropping
- However, FTD is **not forwarding**

Let's review! What are possible reasons that FTD may drop traffic without a Lina syslog or snort verdict indicating a drop?

# Checking Lina inspection and L2 adjacency

Remember, we can use packet capture with the 'trace' command to see policy decisions:

```
firepower# show cap capin trace packet-number 1

7084 packets captured

  1: 13:04:12.548204      802.1Q vlan#36 P0 10.1.1.20.2286
> 10.1.2.100.80: S 1277167793:1277167793(0) win 512
...
Phase: 14
Type: ROUTE-LOOKUP
Subtype: Resolve Egress Interface
Result: ALLOW
Config:
Additional Information:
found next-hop 10.1.2.50 using egress ifc outside
Result:
...
output-interface: outside
...
Action: allow
```

We can see that configured policies are not dropping the packet. However, it is strange that our next hop is **not** the directly-connected server.

Let's investigate this...

# Next-hop ARP resolution?

```
firepower# sh arp | i 10.1.2.50
firepower#
```

Check for ARP entry. Does not exist.

Reason for packet drop:

```
firepower# debug arp
debug arp enabled at level 1
arp-req: generating request for 10.1.2.50 at interface outside
arp-req: request for 10.1.2.50 still pending
```

We can see that ARP resolution is failing for this host. Therefore FTD cannot egress the packet.

Root cause:

```
firepower# show route
...
S      10.1.2.100 255.255.255.255 [1/0] via 10.1.2.50, outside
```

A static, more specific /32 route to the server via 10.1.2.50 is configured and that host is not responding to ARP.



The bridge to possible

Thank you

CISCO *Live!*

CISCO *Live!*

ALL IN