Troubleshoot Traffic Drops Due to LINA Protocol Inspection on FTD

Contents

Introduction

This document describes how to identify if the LINA protocol inspection for Modular Policy Framework (MPF), drops traffic in the Cisco Secure FTD.

Prerequisites

Cisco recommended you have knowledge on these topics:

- Cisco Secure Firewall Threat Defense (FTD).
- Cisco Secure Firewall Manager Center (FMC).

Components Used

The information in this document is based on these software and hardware versions:

- Virtual Cisco Secure Firewall Threat Defense (FTD), version 7.4.2
- Virtual Cisco Secure Firewall Manager Center (FMC), version 7.4.2

The information in this document was created from the devices in a specific lab environment. All the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Background Information

The inspection engines are required in a firewall for services that embed IP addressing information in the user data packet or that open secondary channels on dynamically assigned ports.

The protocol inspection can help prevent malicious traffic from entering the network by inspecting the content of network packets and blocking or modifying traffic based on the application or protocol being used.
As a result, inspection engines can affect overall throughput. Several common inspection engines are enabled on the firewall by default, it can be needed to enable others depending on the network.

Default Configurations

By default, the FTD LINA configuration includes a policy that matches all default application inspection traffic.

The inspection applies to the traffic on all interfaces (a global policy).

Default application inspection traffic includes traffic to the default ports for each protocol. You can only apply one global policy, so if you want to alter the global policy, for example, to apply inspection to non-standard ports, or to add inspections that are not enabled by default, you need to either edit the default policy or disable it and apply a new one.

Run the command **show running-config policy-map** on LINA, FTD Command Line Interface (CLI) via **system support diagnostic-cli**, to get the information.

```
firepower# show running-config policy-map
!
policy-map type inspect dns preset_dns_map
 parameters
  message-length maximum client auto
  message-length maximum 512
  no tcp-inspection
policy-map type inspect ip-options UM_STATIC_IP_OPTIONS_MAP
 parameters
  eool action allow
  nop action allow
  router-alert action allow
policy-map global_policy
 class inspection_default
  inspect dns preset_dns_map
  inspect ftp
  inspect h323 h225
  inspect h323 ras
  inspect rsh
  inspect rtsp
  inspect sqlnet
  inspect skinny
  inspect sunrpc
  inspect sip
  inspect netbios
  inspect tftp
  inspect icmp
  inspect icmp error
  inspect ip-options UM_STATIC_IP_OPTIONS_MAP
```

```
 class class_snmp
  inspect snmp
 class class-default
  set connection advanced-options UM_STATIC_TCP_MAP
!
```

Identify Packet Drops due to the MPF Protocol Inspection

Even when traffic aligns with the Access Control Policy (ACP) assigned to the firewall, in certain scenarios, the inspection process terminates connections due to specific traffic behavior received by the firewall, a not supported design, application standard or an inspection limitation.

During traffic troubleshooting, a useful process to use is:

- Set real-time capture logs on the interfaces from which the traffic flows (ingress and egress interfaces), command:

```
firepower# capture <capture_name> [interface <interface_name>][match <protocol> <source_ip> [port <port
```

Using the captures, you can include the option **packet number X trace detail** and it must provide the result phase by phase the connection takes, as a packet-tracer command does, but with this option you ensure it is real-time traffic.

```
firepower# show capture <capture_name> packet number X trace detail
```

- Set real-time Accelerated Security Path (ASP) Drop, the capture type **asp-drop** shows the packets or connections dropped by the ASP, there is a list of reasons which you can find in the Related links of the document, command:

```
firepower# capture <capture_name> [type <raw-data|asp-drop|asa-dataplane>] [interface <interface_name>]
```

Protocol inspection drops can be ignored, as an **allow** result can be observed in the packet-tracer phases. That is why, it is crucial to always verify the drop reason using real-time capture logs.

Common Drop Error Messages

The Accelerated Security Path (ASP) drop is often used for debugging purposes to help troubleshoot network issues. The **show asp drop** command is utilized to display these dropped packets or connections, providing insights into the reasons for the drops, which can include issues like NAT failures, inspection failures, or access rule denials.

Key Points about ASP drops:

- **Frame Drops**: These are drops related to individual packets, such as invalid encapsulation or no route to host.
- **Flow Drops**: These are related to connections, such as flows denied by access rules or NAT failures.
- **Usage**: The command is primarily for debugging and the output can change.

These error messages or drop reasons are examples you can encounter during troubleshooting. They can defer depending on the Inspection protocol being used.

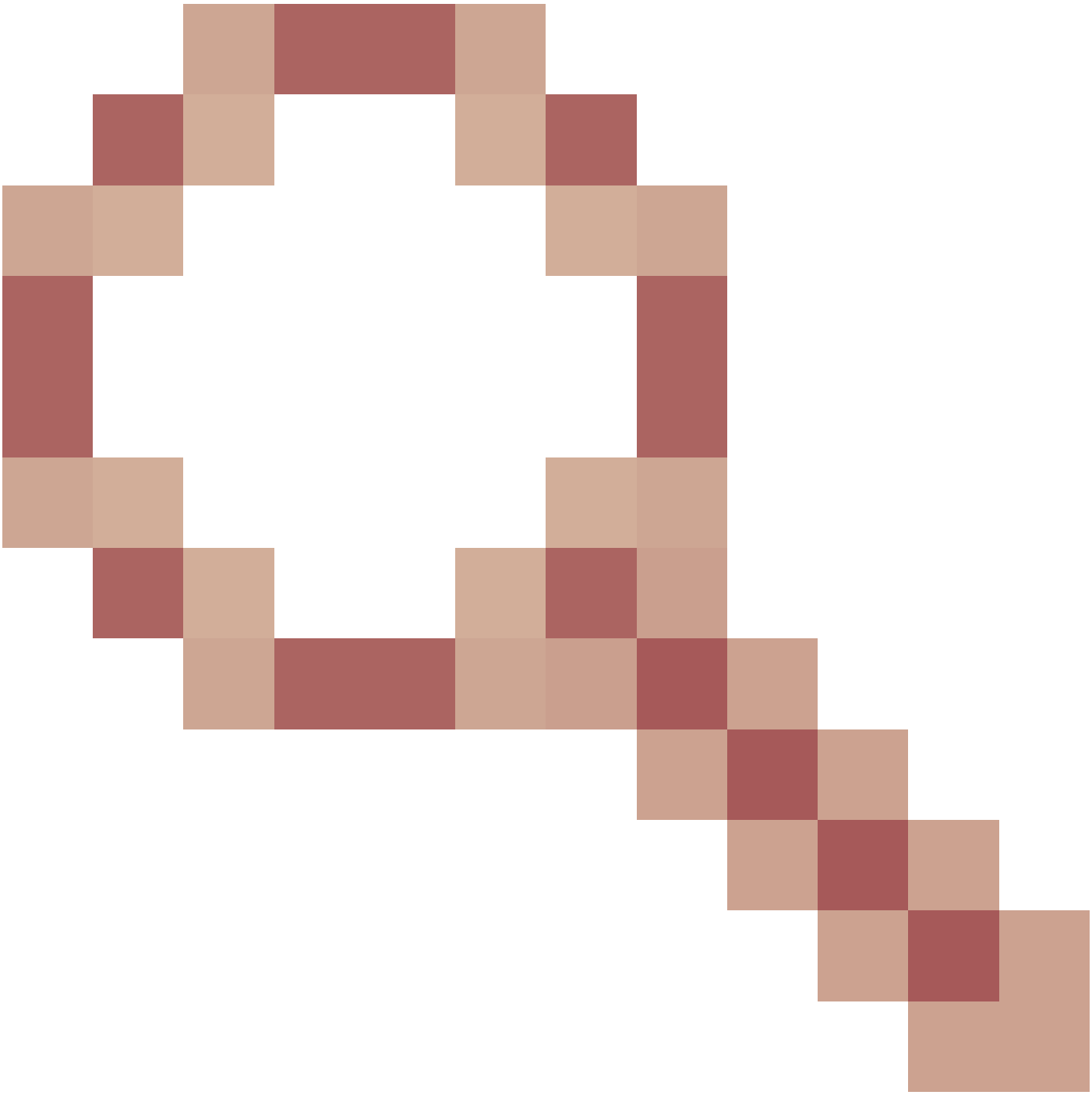
SUN RPC Inspection Drop Example

This scenario is for a single-arm proxy FTDv in AWS deployment, RPC traffic encapsulated by Geneve, if the Sun Rpc inspection is enabled the connection is dropped.

The output shows ASP drops for Sun Rpc inspection, Sun Rcp use port 111 as destination the last packet is Geneve encapsulation port which use 6081 as destination. The drop reason in the output as you can observe is "No valid adjacency"

```
firepower# show capture asp-drop
...
8: 16:23:02.462958 10.0.0.5.780 > 172.16.0.3.111: . ack 526534108 win 29200 Drop-reason: (no-adjacency)
9: 16:23:09.769338 10.0.0.5.780 > 172.16.0.3.111: P 1795131583:1795131679(96) ack 526534108 win 29200 D
10: 16:23:10.148658 172.16.0.3.111 > 10.0.0.5.780: . ack 4026726685 win 26880 Drop-reason: (no-adjacenc
11: 16:23:10.463004 10.0.0.5.780 > 172.16.0.3.111: . ack 526534108 win 29200 Drop-reason: (no-adjacency
12: 16:23:26.462729 10.0.0.5.780 > 172.16.0.3.111: . ack 526534108 win 29200 Drop-reason: (no-adjacency
13: 16:23:27.548692 10.79.67.11.60855 > 10.79.67.4.6081: udp 176 [GENEVE segment-id 0 payload-length 13
```

Cisco bug ID CSCwj00074



FTDv single-arm proxy drops traffic with no-adjacency with inspect **sunrpc** enabled

The traffic is dropped as 'no-valid adjacency' in the ASP of the LINA engine because the source and destination mac address are suddenly populated all in zeros after the second packet (SYN/ACK) of the 3-way handshake.

ASP Drop reason:

Name: **no-adjacency**
No valid adjacency:
This counter increments when the security appliance receives a packet on an existing flow that no longer has a valid output adjacency. This can occur if the next hop is no longer reachable or if a routing change has occurred typically in a dynamic routing environment.

Solution: Disable **sunrpc** inspection.

SQL*NET Inspection Drop Example

This scenario is for a single-arm proxy FTDv in AWS deployment, if Sql*Nel inspection is enabled, the encapsulated traffic by Geneve is dropped.

The output is for the packet captures merged (you can observe the same packet number):
First line: asp-drop packet capture not encapsulated, Sql*Net use 1521 port as destination.
Second line: VNI interface asp-drop on LINA, Geneve use encapsulation port 6081 as destination.

There are two different drop reasons in the output, as you can observe they are "tcp-buffer-timeout" and "tcp-not-syn"

```
95    2024-12-14 07:55:58.771764    172.16.0.14    10.0.8.2    TCP    251    53905 → 1521 [PSH, ACK] Seq=
95: 07:55:58.771764       10.7.0.3.64056 > 10.7.2.5.6081:  udp 209 [GENEVE segment-id 0 payload-length 169] Drop-

96    2024-12-14 07:55:58.771780    172.16.0.14    10.0.8.2    TCP    1514    [TCP Out-Of-Order] 53905 → 1521 [AC
96: 07:55:58.771780       10.7.0.3.64056 > 10.7.2.5.6081:  udp 1472 [GENEVE segment-id 0 payload-length 1432] Dro

99    2024-12-14 07:55:58.997049    172.16.0.14    10.0.8.2    TCP    308    53903 → 1521 [PSH, ACK] Seq=1 Ack=1
99: 07:55:58.997049       10.7.0.3.64056 > 10.7.2.5.6081:  udp 266 [GENEVE segment-id 0 payload-length 226] Drop-

100    2024-12-14 07:55:58.997079    172.16.0.14    10.0.8.2    TCP    1514    [TCP Out-Of-Order] 53903 → 1521 [A
100: 07:55:58.997079       10.7.0.3.64056 > 10.7.2.5.6081:  udp 1472 [GENEVE segment-id 0 payload-length 1432] Dr
```

ASP Drop reason:

Name: **tcp-buffer-timeout**
TCP Out-of-Order packet buffer timeout:
This counter is incremented and the packet is dropped when a queued out of order TCP packet has been held in the buffer for too long.Typically, TCP packets are put into order on connections that are inspected by the security appliance or when packets are sent to the SSM for inspection. When the next expected TCP packet does not arrive within a certain period, the queued out of order packet is dropped.

Recommendations:
The next expected TCP packet do not arrive due to congestion in the network which is normal in a busy network. The TCP retransmission mechanism in the end host must retransmit the packet and the session can continue.

Name: **tcp-not-syn**
First TCP packet not SYN:
Received a non SYN packet as the first packet of a non intercepted and non nailed connection.

Recommendation:
Under normal conditions, this can be seen when the appliance has already closed a connection, and the client or server still believe the connection is open, and continue to transmit data. Some examples where this can occur is just after a 'clear local-host' or 'clear xlate' is issued. Also, if connections have not been recently removed, and the counter is incrementing rapidly, the appliance can be under attack. Capture a sniffer trace to help isolate the cause.

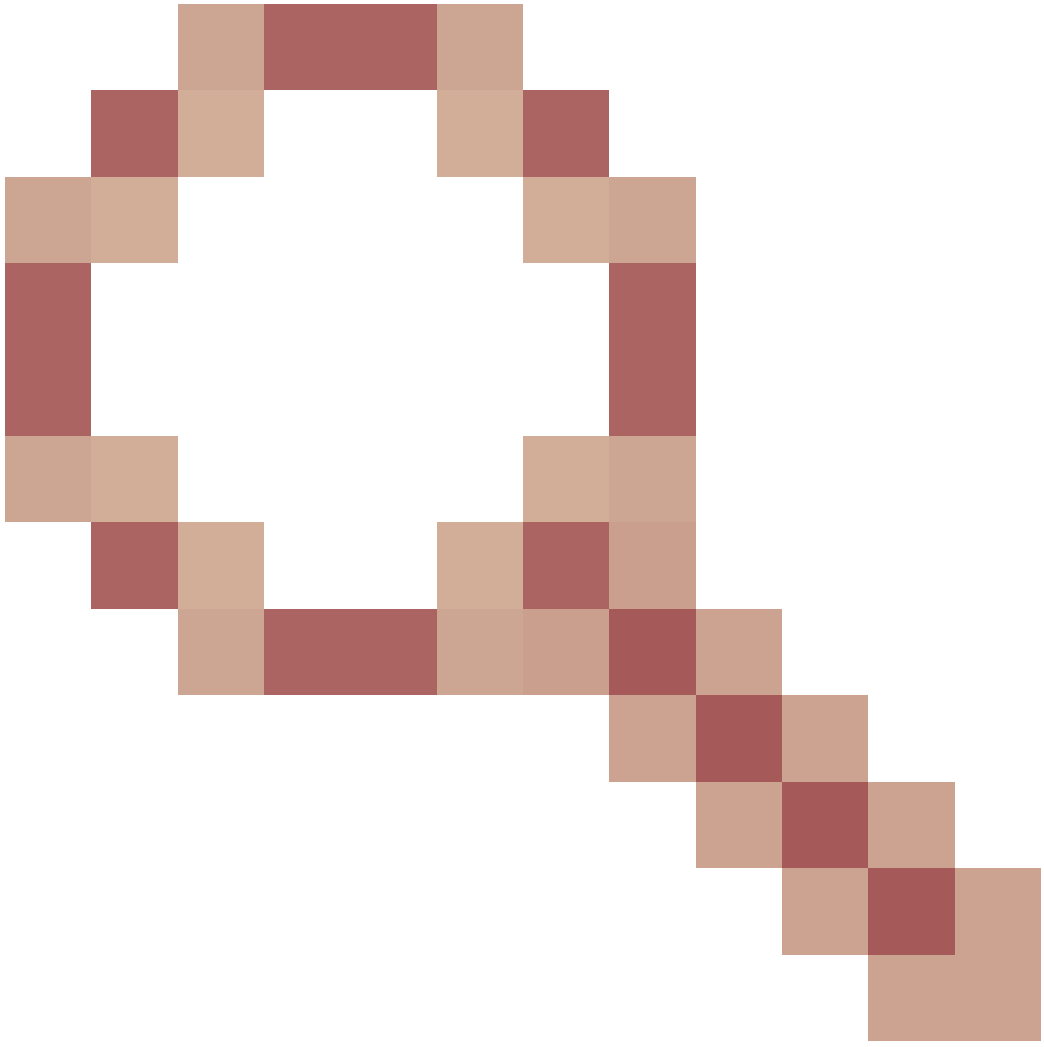
Solution: Disable SQL*Net inspection when SQL data transfer occurs on the same port as the SQL control

TCP port 1521. The security appliance acts as a proxy when SQL*Net inspection is enabled and reduces the client window size from 65000 to about 16000 causing data transfer issues.

ICMP Inspection Drop Example

This scenario is for a FTD cluster environment.
The ICMP identifier of ICMP header can be used as the source port of the 5-tuple in the flow, so all the 5-tuple of the ping packets are the same, ASP drop reason is "inspect-icmp-seq-num-not-matched" as you can observe in this output.

```
firepower#show cap asp-drop
 1: 19:47:09.293136 10.0.5.8 > 10.50.0.53 icmp: echo reply Drop-reason: (inspect-icmp-seq-num-not-match
```



Cisco bug ID CSCvb92417

Cluster ASA drops to-the-box ICMP replies with reason "inspect-icmp-seq-num-not-matched"

ASP Drop reason:

Name: **inspect-icmp-seq-num-not-matched**
ICMP Inspect seq num not matched:
    This counter must increment when the sequence number in the ICMP echo reply message does not match any ICMP echo message that passed across the appliance earlier on the same connection.

Solution: Disable ICMP Inspection. In cluster environment: two or more FTD in cluster and the ICMP traffic can be asymmetric. It is observed that there is a delay for ICMP flow deletion, the subsequent ping is sent fast before the previous ping flow had been cleaned up. In this case consecutive ping packet lost can happen.

SIP Inspection Drop Example

In this scenario, the calls lasted only five minutes, then the connection is drop. When RTP is used, SIP Inspection can drop connections.
As you can observe in the packet capture output on interface for VoIP traffic, the BYE flag in the SIP traffic means the phone call is closed at that time.

```
1    2023-10-13 18:39:03.421456    10.6.6.66    172.16.3.77    SIP/SDP    1055    Request: INVITE sip:1
2    2023-10-13 18:39:03.448325    172.16.3.77    10.6.6.66    SIP    497    Status: 100 Trying |
3    2023-10-13 18:39:03.525424    172.16.3.77    10.6.6.66    SIP    687    Status: 401 Unauthorized |
4    2023-10-13 18:39:03.525943    10.6.6.66    172.16.3.77    SIP    425    Request: ACK sip:123456789
5    2023-10-13 18:39:03.527331    10.6.6.66    172.16.3.77    SIP/SDP    1343    Request: INVITE sip:1
6    2023-10-13 18:39:03.553544    172.16.3.77    10.6.6.66    SIP    497    Status: 100 Trying |
7    2023-10-13 18:39:05.902815    172.16.3.77    10.6.6.66    SIP/SDP    992    Status: 183 Session Pr
8    2023-10-13 18:39:06.091822    172.16.3.77    10.6.6.66    SIP/SDP    967    Status: 180 Ringing |
9    2023-10-13 18:39:13.114435    172.16.3.77    10.6.6.66    SIP/SDP    1063    Status: 200 OK (INVIT
10    2023-10-13 18:39:13.115899    10.6.6.66    172.16.3.77    SIP    560    Request: ACK sip:55663399
11    2023-10-13 18:40:29.206593    172.16.3.77    10.6.6.66    SIP    642    Request: UPDATE sip:FD3a5
12    2023-10-13 18:40:29.207630    10.6.6.66    172.16.3.77    SIP    659    Status: 200 OK (UPDATE) |
13    2023-10-13 18:41:09.940854    10.6.6.66    172.16.3.77    SIP    684    Request: BYE sip:33445566
14    2023-10-13 18:41:10.003066    172.16.3.77    10.6.6.66    SIP    659    Status: 200 OK (BYE) |
```

In this other example, the syslog shows a mapped IP which use PAT, the IP is left with only one port available and the SIP session landed on same port, SIP failed due to port allocation. If PAT is in use SIP inspection can drop the connection.

The ASP drop reason is: "Unable to create UDP connection from IP/port to IP/port due to reaching per-host PAT port block limit of X" and "terminated by inspection engine, reason - reset based on 'service resetinboud' configuration"

```
Nov 18 2019 10:19:34: %FTD-6-607001: Pre-allocate SIP Via UDP secondary channel for 3111:10.11.0.13/506
Nov 18 2019 10:19:35: %FTD-6-302022: Built backup stub TCP connection for identity:172.16.2.20/2325 (17
Nov 18 2019 10:19:38: %FTD-3-305016: Unable to create UDP connection from 3111:10.11.0.12/50195 to 3121
Nov 18 2019 10:19:38: %FTD-4-507003: udp flow from 3111:10.11.0.12/5060 to 3121:10.21.0.12/5060 termina
Nov 18 2019 10:19:39: %FTD-3-305016: Unable to create UDP connection from 3111:10.11.0.12/50195 to 3121
Nov 18 2019 10:19:39: %FTD-4-507003: udp flow from 3111:10.11.0.12/5060 to 3121:10.21.0.12/5060 termina
```

ASP Drop reason:

Name: **async-lock-queue-limit**
Async lock queue limit exceeded:
Each async lock working queue has a limit of 1000. When more SIP packets are attempted to be dispatch to the work queue, packet must be dropped.
Recommendation:
Only SIP traffic can be dropped. When SIP packets have the same parent lock and they can be queued

into the same async lock queue, thus can result into blocks depletion, because only single core is handling all the media. If a SIP packet attempts to be queued when the size of the async lock queue exceeds the limit, the packet must be dropped.

Name: **sp-looping-address**
looping-address:
This counter is incremented when the source and destination addresses in a flow are the same. SIP flows where address privacy is enabled are excluded, as it is normal for those flows to have the same source and destination address.

Recommendation:
There are two possible conditions when this counter can increment. One is when the appliance receives a packet with the source address equal to the destination. This represents a type of DoS attack. The second is when the NAT configuration of the appliance NATs a source address to equal that of the destination.

Name: **parent-closed**
Parent flow is closed:
When the parent flow of a subordinating flow is closed, the subordinating flow is also closed. For example, an FTP data flow (subordinating flow) can be closed with this specific reason when its control flow (parent flow) is terminated. This reason is also given when a secondary flow (pin-hole) is closed by its controlling application. For example, when the BYE messaged is received, the SIP inspection engine (controlling application) must close the corresponding SIP RTP flows (secondary flow).

Solution: Disable SIP Inspection. Due to limitations on the protocol:

- SIP inspection supports the Chat feature only. Whiteboard, File Transfer, and Application Sharing are not supported. RTC Client 5.0 is not supported.
- When using PAT, any SIP header field which contains an internal IP address without a port can not be translated and hence the internal IP address can be leaked outside. If you want to avoid this leakage, configure NAT instead of PAT.
- SIP inspection is enabled by default using the default inspection map, which includes:
  * SIP instant messaging (IM) extensions: Enabled.
  * Non-SIP traffic on SIP port: Dropped.
  * Hide server and endpoint IP addresses: Disabled.
  * Mask software version and non-SIP URIs: Disabled.
  * Ensure that the number of hops to destination is greater than 0: Enabled.
  * RTP conformance: Not enforced.
  * SIP conformance: Do not perform state checking and header validation.

Troubleshoot

These are some of the suggested commands to troubleshoot traffic issues related to the LINA MPF protocol inspection.

- **Show service-policy** display the service policy statistics for the LINA MPF inspections enabled.

```
firepower# show service-policy

Global policy:
  Service-policy: global_policy
    Class-map: inspection_default
      Inspect: dns preset_dns_map, packet 0, lock fail 0, drop 0, reset-drop 0, 5-min-pkt-rate 0 pkts/s
```

```
      Inspect: ftp, packet 0, lock fail 0, drop 0, reset-drop 0, 5-min-pkt-rate 0 pkts/sec, v6-fail-clo
      Inspect: h323 h225 _default_h323_map, packet 0, lock fail 0, drop 0, reset-drop 0, 5-min-pkt-rate
               tcp-proxy: bytes in buffer 0, bytes dropped 0
      Inspect: h323 ras _default_h323_map, packet 0, lock fail 0, drop 0, reset-drop 0, 5-min-pkt-rate
      Inspect: rsh, packet 0, lock fail 0, drop 0, reset-drop 0, 5-min-pkt-rate 0 pkts/sec, v6-fail-clo
      Inspect: rtsp, packet 0, lock fail 0, drop 0, reset-drop 0, 5-min-pkt-rate 0 pkts/sec, v6-fail-cl
               tcp-proxy: bytes in buffer 0, bytes dropped 0
      Inspect: sqlnet, packet 0, lock fail 0, drop 0, reset-drop 0, 5-min-pkt-rate 0 pkts/sec, v6-fail-
      Inspect: skinny, packet 0, lock fail 0, drop 0, reset-drop 0, 5-min-pkt-rate 0 pkts/sec, v6-fail-
               tcp-proxy: bytes in buffer 0, bytes dropped 0
      Inspect: sunrpc, packet 0, lock fail 0, drop 0, reset-drop 0, 5-min-pkt-rate 0 pkts/sec, v6-fail-
               tcp-proxy: bytes in buffer 0, bytes dropped 0
      Inspect: sip , packet 0, lock fail 0, drop 0, reset-drop 0, 5-min-pkt-rate 0 pkts/sec, v6-fail-cl
               tcp-proxy: bytes in buffer 0, bytes dropped 0
      Inspect: netbios, packet 0, lock fail 0, drop 0, reset-drop 0, 5-min-pkt-rate 0 pkts/sec, v6-fail
      Inspect: tftp, packet 0, lock fail 0, drop 0, reset-drop 0, 5-min-pkt-rate 0 pkts/sec, v6-fail-cl
      Inspect: icmp, packet 0, lock fail 0, drop 0, reset-drop 0, 5-min-pkt-rate 0 pkts/sec, v6-fail-cl
      Inspect: icmp error, packet 0, lock fail 0, drop 0, reset-drop 0, 5-min-pkt-rate 0 pkts/sec, v6-f
      Inspect: ip-options UM_STATIC_IP_OPTIONS_MAP, packet 0, lock fail 0, drop 0, reset-drop 0, 5-min-
    Class-map: class_snmp
      Inspect: snmp, packet 0, lock fail 0, drop 0, reset-drop 0, 5-min-pkt-rate 0 pkts/sec, v6-fail-cl
    Class-map: class-default

      Default Queueing      Set connection policy:         drop 0
      Set connection advanced-options: UM_STATIC_TCP_MAP
        Retransmission drops: 0                   TCP checksum drops : 0
        Exceeded MSS drops  : 0                   SYN with data drops: 0
        Invalid ACK drops   : 0                   SYN-ACK with data drops: 0
        Out-of-order (OoO) packets : 0            OoO no buffer drops: 0
        OoO buffer timeout drops : 0              SEQ past window drops: 0
        Reserved bit cleared: 0                   Reserved bit drops : 0
        IP TTL modified     : 0                   Urgent flag cleared: 0
        Window varied resets: 0
        TCP-options:
          Selective ACK cleared: 0                Timestamp cleared  : 0
          Window scale cleared : 0
          Other options cleared: 0
          Other options drops: 0
```

This sample output from the show service-policy inspect http command shows the http statistics:

```
firepower# show service-policy inspect http
    Global policy:
      Service-policy: global_policy
        Class-map: inspection_default
          Inspect: http http, packet 1916, drop 0, reset-drop 0
            protocol violations
              packet 0
            class http_any (match-any)
              Match: request method get, 638 packets
              Match: request method put, 10 packets
              Match: request method post, 0 packets
              Match: request method connect, 0 packets
              log, packet 648
```

- Set an **asp-drop** capture on the interface to inspect.

```
Syntax
#Capture <capture name> type asp-drop <reason> match <protocol> <source> <destination>

for example
#Capture asp type asp-drop all match ip any any
#Capture asp type asp-drop all match ip any host x.x.x.x
#Capture asp type asp-drop all match ip host x.x.x.x host x.x.x.x
```

How to Enable or Disable Specific LINA MPF Application Inspections

These are the available options to enable or disable the MPF LINA application inspections in the Cisco Secure Firewall Threat Defense.

- Configuration over FlexConfig: You need admin access to the FMC UI, this change is permanent on the configuration.
- Configuration over FTD CLI: You need admin access to FTD CLI, this change is not permanent, if a reboot or a new deployment takes place, the configuration is removed.

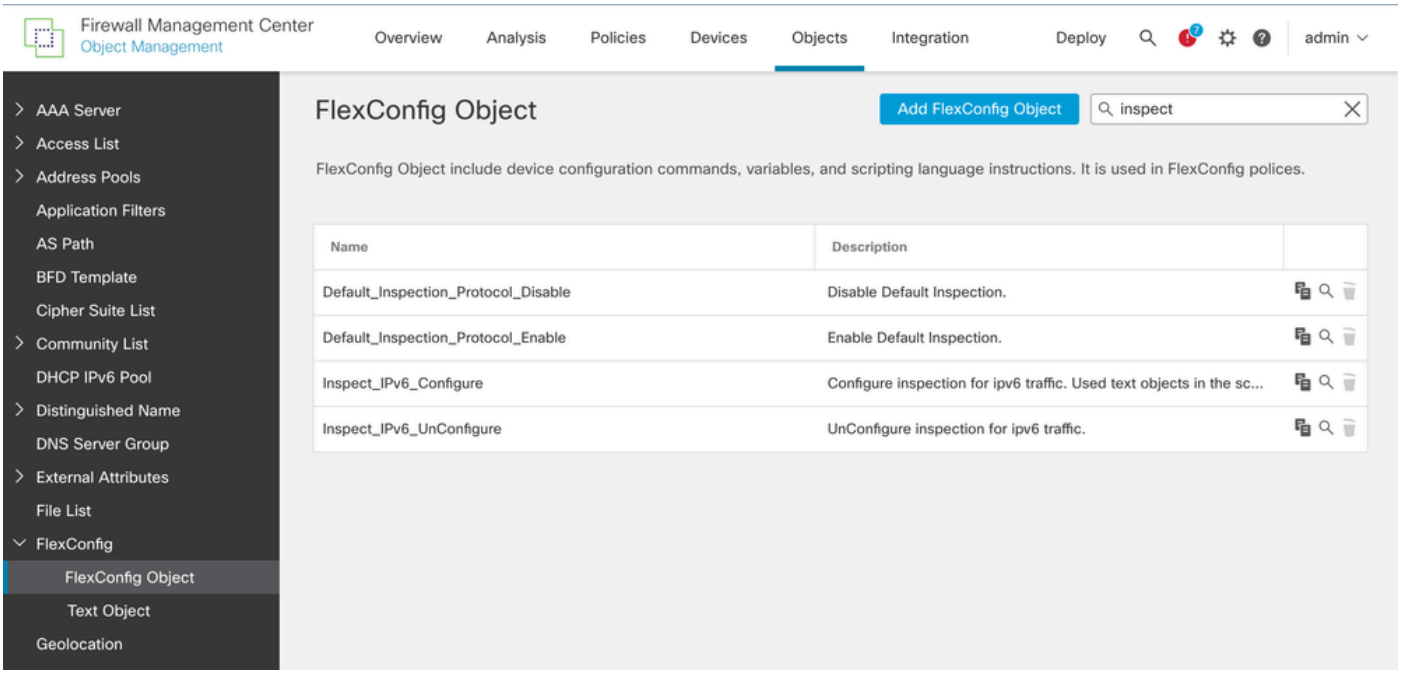
Configuration over FlexConfig

FlexConfig is a method of last resort to configure ASA-based features that are compatible with threat defense but which are not otherwise configurable in management center.

The configuration to disable or enable inspection permanently is on FlexConfig over the FMC UI, it can be applied globally or for specific traffic only.

Step 1.

On FMC UI, navigate to **Objects > Object Management > FlexConfig > FlexConfig Object**, there you can find the list of the default Protocol Inspection objects.



Step 2.

To disable a specific protocol inspection, you can create a FlexConfig Object.

Navigate to **Objects > Object Management > FlexConfig > FlexConfig Object > Add FlexConfig Object**

In this example, the configuration to disable SIP Inspection from the global_policy, the syntax must be:

```
policy-map global_policy
 class inspection_default
  no inspect sip
```

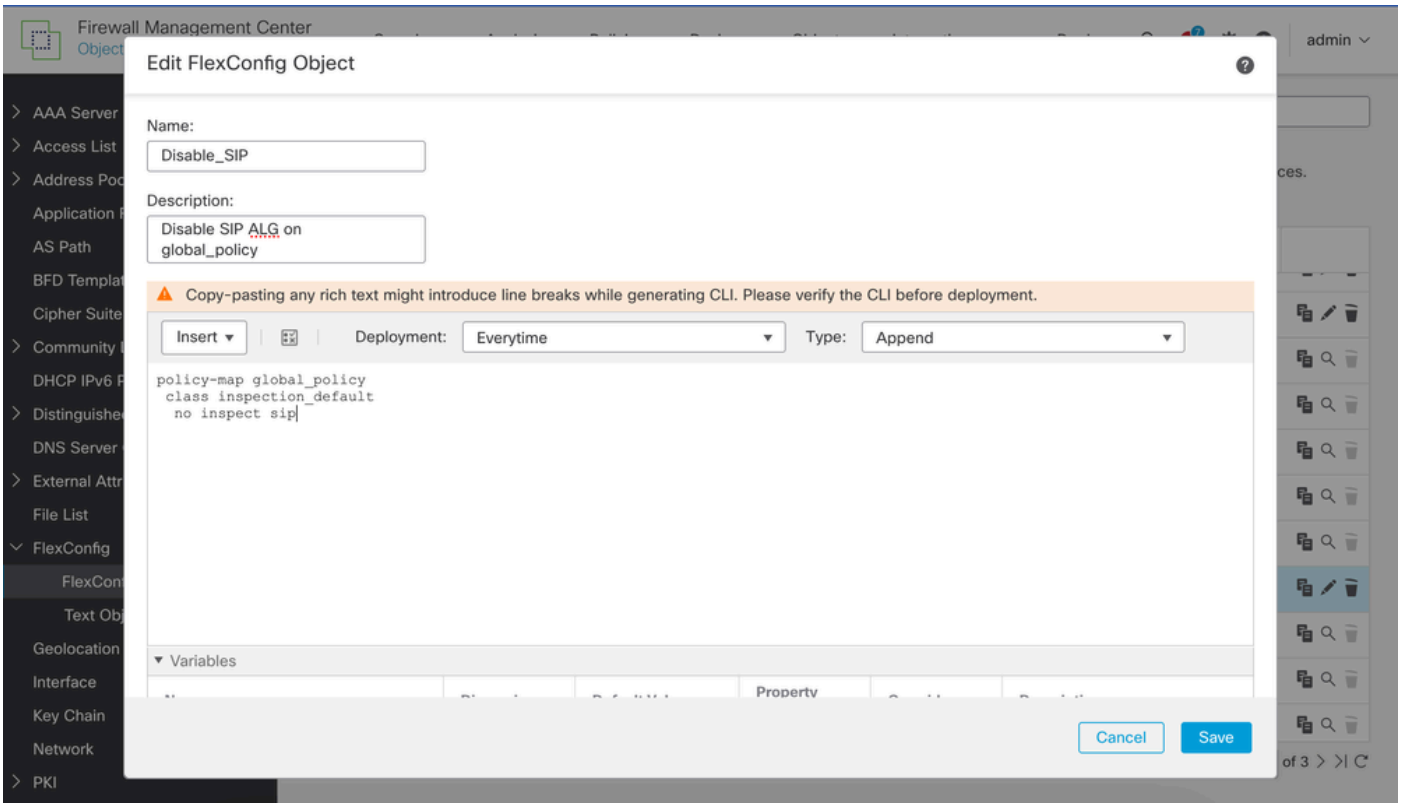
When configuring a FlexConfig object, you can choose deployment frequency and type.

Deployment

- If the FlexConfig object points to system-managed objects such as network or ACL objects, choose Everytime. Otherwise, updates to the objects could not get deployed.
- Use Once if the only thing you do in the object is to clear a configuration. Then, remove the object from the FlexConfig policy after the next deployment.

Type

- Append (The default.) Commands in the object are put at the end of the configurations generated from the management center policies. You must use Append if you use policy object variables, which point to objects generated from managed objects. If commands generated for other policies overlap with those specified in the object, you must select this option so your commands are not overwritten. This is the safest option.
- Prepend. Commands in the object are put at the beginning of the configurations generated from the management center policies. You would typically use prepend for commands that clear or negate a configuration.



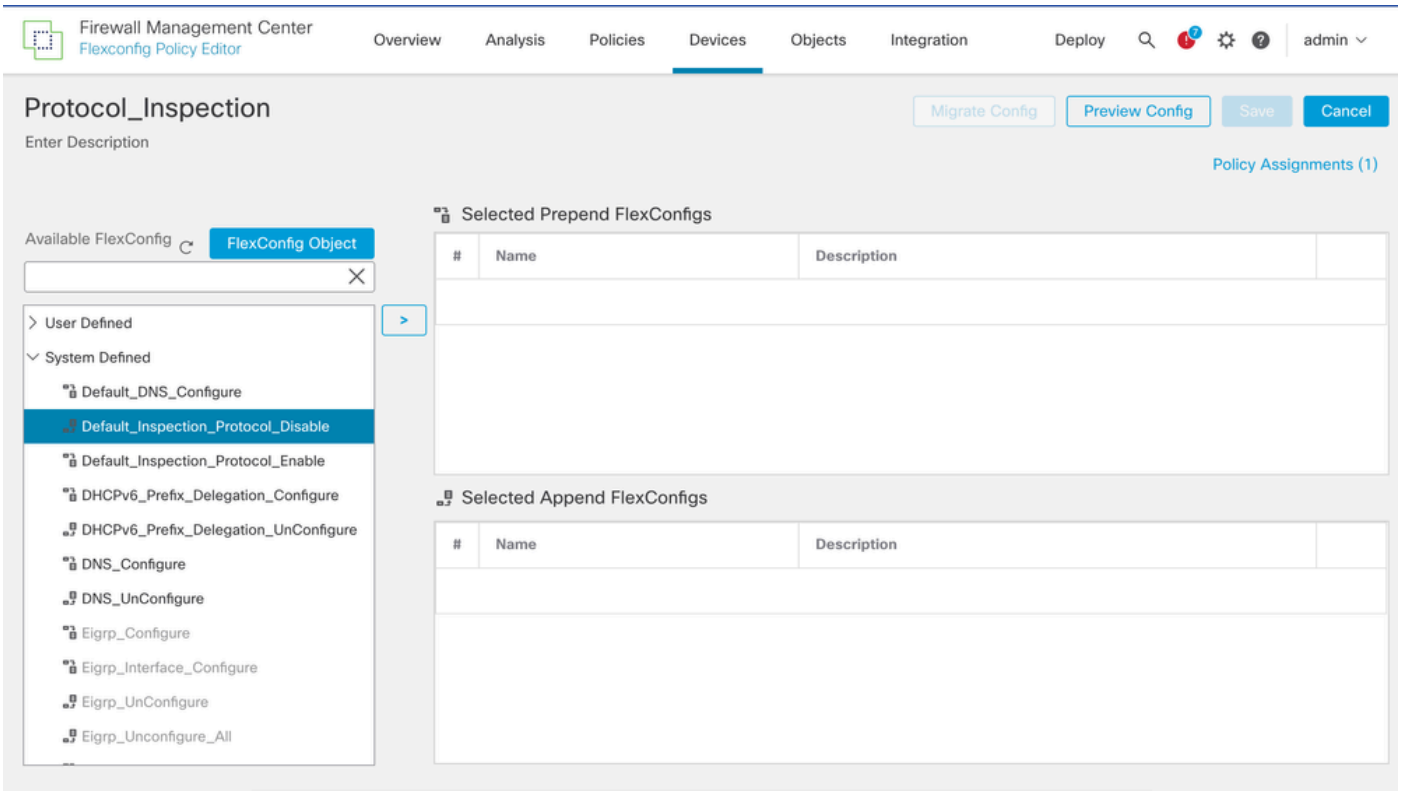
Create an object to disable a single protocol from the default global_policy

Step 3.

Add the objects in the FlexConfig policy assigned to LINA.

Navigate to **Devices > FlexConfig** select the FlexConfig policy applied to the firewall with drop issues.

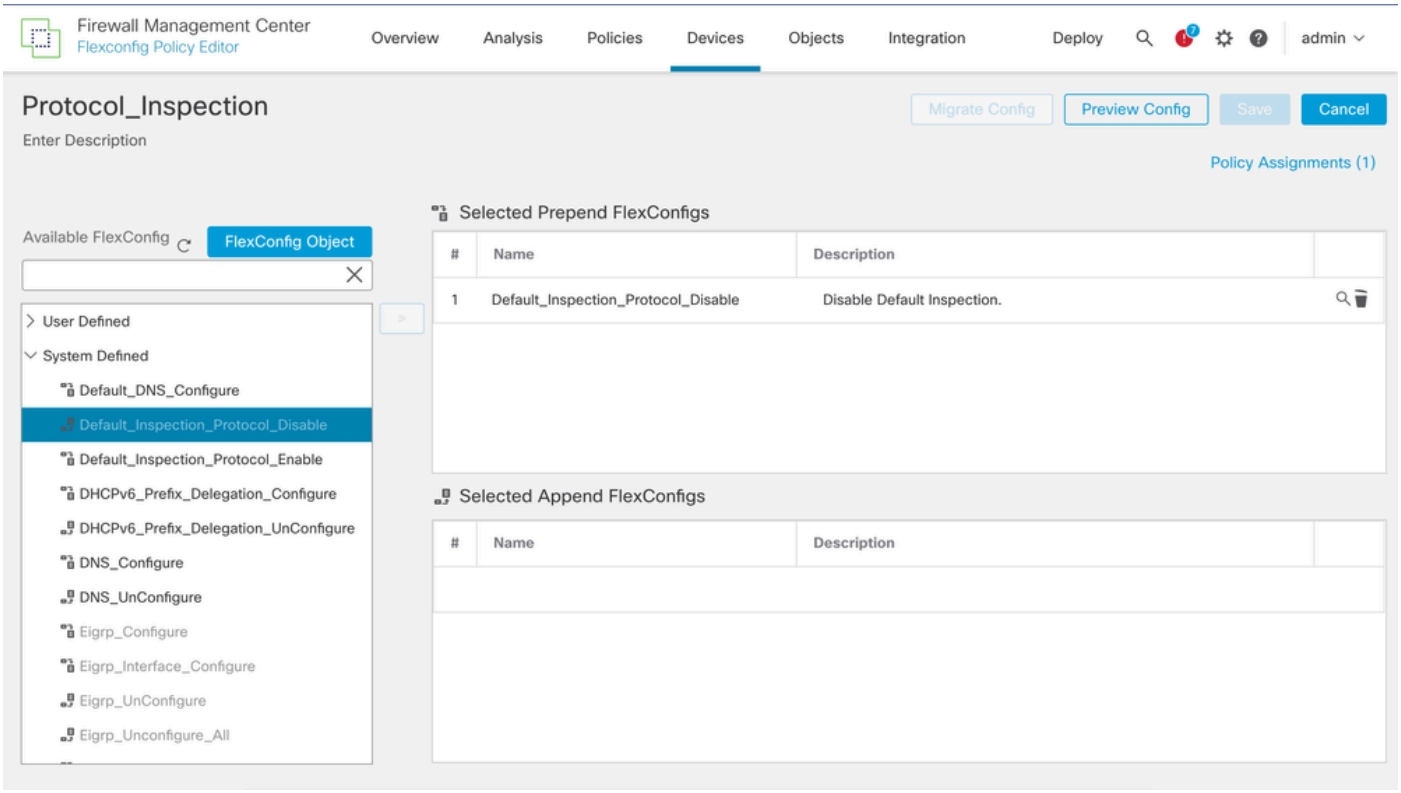
To disable all inspection globally, select the Object **Default_Inspection_Protocol_Disable** under the System Defined FlexConfig Objects, then click on the blue arrow in between to add it to the FlexConfig Policy.



Select the System defined Object to Disable all Protocol Inspection

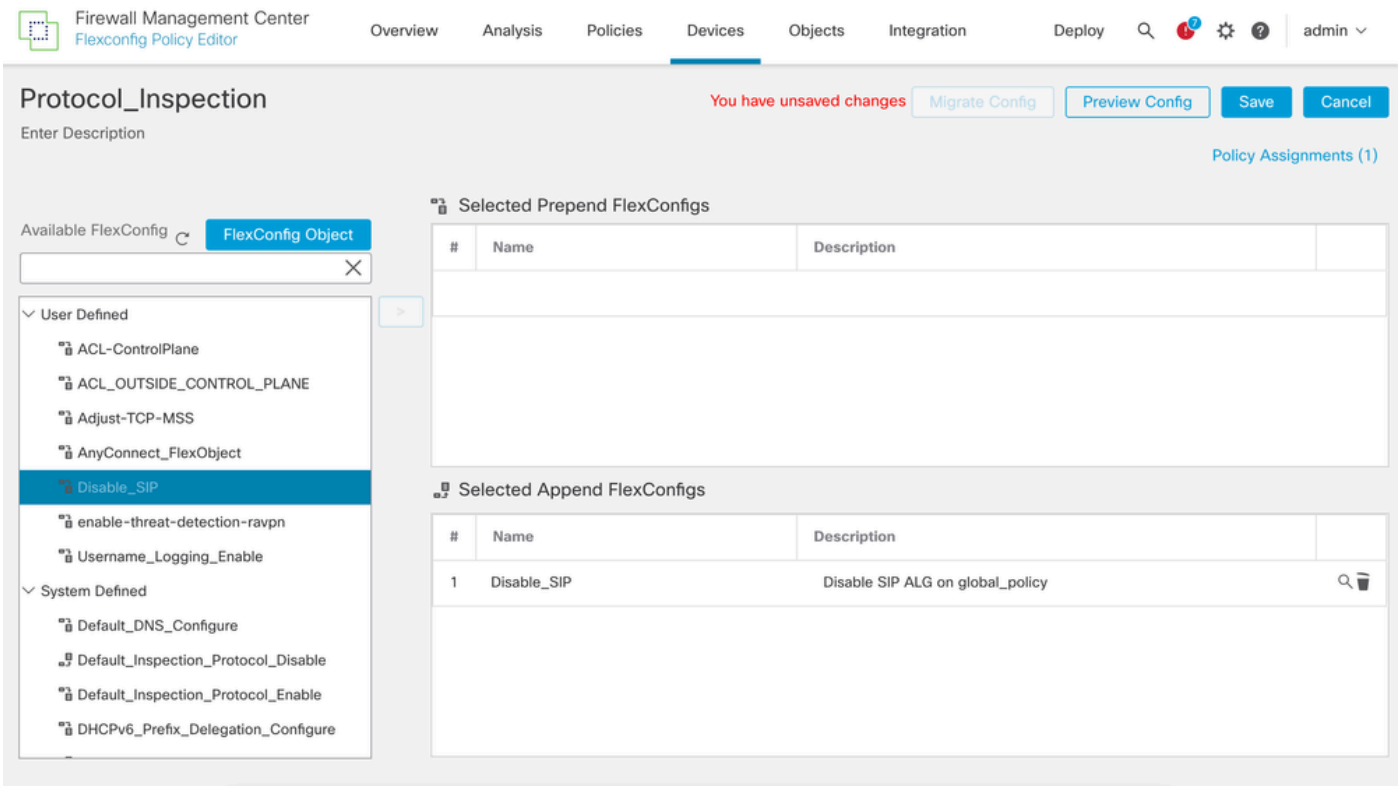
Step 4.

Once selected, confirm it appears in the right boxes, do not forget to **save** and **deploy** the configuration to take effect.



Selected Object to Disable all Protocol Inspection

Step 5.
To disable a single protocol inspection, select the object previously created from User defined list and add it to the policy using the arrow in between the boxes.



Select to Disable a Single Protocol Inspection from the global_policy

Step 6.

Once selected, confirm it appears in the right boxes, do not forget to **save** and **deploy** the configuration to take effect.

Configuration using the FTD CLI

This solution can be applied immediately from the FTD CLI to test if inspection is affecting the traffic. However, the configuration change is not saved if a reboot or a new deployment occurs.

The command must be executed from the FTD CLI in Clish mode.

```
> configure inspection <inspection_type> disable

for example
> configure inspection SIP disable
```

Verify

To verify that the protocol disable is effective, execute the command **show running-config policy-map**. In this example, SIP inspection is disabled as it no longer appears in the default protocol list.

```
firepower# show running-config policy-map
!
policy-map type inspect dns preset_dns_map
 parameters
  message-length maximum client auto
  message-length maximum 512
  no tcp-inspection
policy-map type inspect ip-options UM_STATIC_IP_OPTIONS_MAP
 parameters
  eool action allow
  nop action allow
  router-alert action allow
policy-map global_policy
 class inspection_default
  inspect dns preset_dns_map
  inspect ftp
  inspect h323 h225
  inspect h323 ras
  inspect rsh
  inspect rtsp
  inspect sqlnet
  inspect skinny
  inspect sunrpc
  inspect netbios
  inspect tftp
  inspect icmp
  inspect icmp error
  inspect ip-options UM_STATIC_IP_OPTIONS_MAP
 class class_snmp
  inspect snmp
 class class-default
  set connection advanced-options UM_STATIC_TCP_MAP
!
firepower#
```

Related Information

Technical Support & Documentation - Cisco Systems

- Getting Started with Application Layer Protocol Inspection
- Inspection of Basic Internet Protocols
- Show ASP Drop Command Usage