Configuring ERSPAN

This chapter describes how to configure an encapsulated remote switched port analyzer (ERSPAN) to transport mirrored traffic in an IP network on Cisco NX-OS devices.

This chapter contains the following sections:

About ERSPAN

ERSPAN transports mirrored traffic over an IP network, which provides remote monitoring of multiple switches across your network. The traffic is encapsulated at the source router and is transferred across the network. The packet is decapsulated at the destination router and then sent to the destination interface.

ERSPAN Sources

The interfaces from which traffic can be monitored are called ERSPAN sources. Sources designate the traffic to monitor and whether to copy ingress, egress, or both directions of traffic. ERSPAN sources include the following:

- Ethernet ports (but not subinterfaces)
- Port channels
- Forward drops

Note A single ERSPAN session can include mixed sources in any combination of the above.

ERSPAN Sessions

You can create ERSPAN sessions that designate sources to monitor.

Localized ERSPAN Sessions

An ERSPAN session is localized when all of the source interfaces are on the same line card.

Licensing Requirements for ERSPAN

The following table shows the licensing requirements for this feature:

Product	License Requirement
Cisco NX-OS	ERSPAN requires no license. Any feature not included in a license package is bundled with the nx-os is provided at no extra charge to you. For a complete explanation of the Cisco NX-OS licensing schem Cisco NX-OS Licensing Guide.

Prerequisites for ERSPAN

ERSPAN has the following prerequisites:

- You must first configure the ports on each device to support the desired ERSPAN configuration. For more information, see the *Cisco Nexus 3400-S Series NX-OS Interfaces Configuration Guide*.

Guidelines and Limitations for ERSPAN

ERSPAN has the following configuration guidelines and limitations:

- ERSPAN truncation is not supported on Cisco Nexus 3400 Series switches.
- For ERSPAN session limits, see the *Cisco Nexus 3400 Series NX-OS Verified Scalability Guide*.
- Two ERSPAN destination sessions are not supported on Cisco Nexus 3400-S platform switches.
- Only ERSPAN source sessions are supported. Destination sessions are not supported.
- ERSPAN destination as a Port channel is not supported.
- Statistics are not supported for the filter access group.
- An access-group filter in an ERSPAN session must be configured as vlan-accessmap.
- Control plane packets that are generated by the supervisor cannot be ERSPAN encapsulated or filtered by an ERSPAN access control list (ACL).
- ERSPAN is not supported for management ports.
- ERSPAN does not support destinations on Layer 3 port-channel subinterfaces.
- Configuring UDF based filter is supported only on Ethernet ports and Port-channels.

- If you enable ERSPAN on a vPC and ERSPAN packets must be routed to the destination through the vPC, packets that come through the vPC peer link cannot be captured.
- For SPAN forward drop traffic, SPAN only the packets that get dropped due to various reasons in the forwarding plane. This enhancement is supported only for ERSPAN Source session. It is not supported along with SPAN ACL and source interface.
- ERSPAN is not supported over a VXLAN overlay.
- ERSPAN works on default and nondefault VRFs.

The following guidelines and limitations apply to egress (Tx) ERSPAN:

- The flows for post-routed unknown unicast flooded packets are in the ERSPAN session, even if the ERSPAN session is configured to not monitor the ports on which this flow is forwarded.

Default Settings

The following table lists the default settings for ERSPAN parameters.

Table 1: Default ERSPAN Parameters

Parameters	Default
ERSPAN sessions	Created in the shut state
ERSPAN marker packet interval	100 microseconds
Timestamp granularity of ERSPAN Type III sessions	100 picoseconds

Configuring ERSPAN

Configuring an ERSPAN Source Session

You can configure an ERSPAN session on the local device only. By default, ERSPAN sessions are created in the shut state.

Note ERSPAN does not monitor any packets that are generated by the supervisor, regardless of their source.

SUMMARY STEPS

1. **configure terminal**
2. **monitor erspan origin ip-address** *ip-address* **global**
3. **no monitor session** {*session-number* | **all**}
4. **monitor session** {*session-number* | **all**} **type erspan-source** [**shut**]
5. **description** *description*

6. **source** {**interface** *type* [**tx** | **rx** |**both**] }
7. (Optional) Repeat Step 7 to configure all ERSPAN sources.
8. **destination ip** *ip-address*
9. **erspan-id** *erspan-id*
10. **vrf** *vrf-name*
11. (Optional) **ip ttl** *ttl-number*
12. (Optional) **ip dscp** *dscp-number*
13. **no shut**
14. **exit**
15. (Optional) **show monitor session** {**all** | *session-number* | **range** *session-range*} [**brief**]
16. (Optional) **show running-config monitor**
17. (Optional) **show startup-config monitor**
18. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**monitor erspan origin ip-address** *ip-address* **global** **Example:** `switch(config)# monitor erspan origin ip-address 10.0.0.1 global`	Configures the ERSPAN global origin IP address.

Step 3	**no monitor session** {*session-number* | **all**}
	Example:
	`switch(config)# no monitor session 3`
		Clears the configuration of the specified ERSPAN session. The new session configuration is added to the existing session configuration.

Step 4	**monitor session** {*session-number* | **all**} **type erspan-source** [**shut**]
	Example:
	`switch(config)# monitor session 3 type erspan-source`
	`switch(config-erspan-src)#`
		Configures an ERSPAN Type II source session. By default the session is bidirectional. The optional keyword shut specifies a shut state for the selected session.

Step 5	**description** *description* **Example:** `switch(config-erspan-src)# description erspan_src_session_3`	Configures a description for the session. By default, no description is defined. The description can be up to 32 alphanumeric characters.

Step 6	**source** {**interface** *type* [**tx** | **rx** |**both**] }
	Example:
	`switch(config-erspan-src)# source interface ethernet 2/1-3, ethernet 3/1 rx`
		You can configure one or more sources, as either a series of comma-separated entries or a range of numbers. You can specify the traffic direction to copy as ingress, egress, or both.

	Command or Action	Purpose
	Example: `switch(config-erspan-src)# source interface port-channel 2`	For a unidirectional session, the direction of the source must match the direction specified in the session.
Step 7	(Optional) Repeat Step 7 to configure all ERSPAN sources.	—
Step 8	**destination ip** *ip-address* **Example:** `switch(config-erspan-src)# destination ip 10.1.1.1`	Configures the destination IP address in the ERSPAN session. Only one destination IP address is supported per ERSPAN source session.
Step 9	**erspan-id** *erspan-id* **Example:** `switch(config-erspan-src)# erspan-id 5`	Configures the ERSPAN ID for the ERSPAN source session. The ERSPAN range is from 1 to 1023.
Step 10	**vrf** *vrf-name* **Example:** `switch(config-erspan-src)# vrf default`	Configures the virtual routing and forwarding (VRF) instance that the ERSPAN source session uses for traffic forwarding. The VRF name can be any case-sensitive, alphanumeric string up to 32 characters.
Step 11	(Optional) **ip ttl** *ttl-number* **Example:** `switch(config-erspan-src)# ip ttl 25`	Configures the IP time-to-live (TTL) value for the ERSPAN traffic. The range is from 1 to 255.
Step 12	(Optional) **ip dscp** *dscp-number* **Example:** `switch(config-erspan-src)# ip dscp 42`	Configures the differentiated services code point (DSCP) value of the packets in the ERSPAN traffic. The range is from 0 to 63.
Step 13	**no shut** **Example:** `switch(config-erspan-src)# no shut`	Enables the ERSPAN source session. By default, the session is created in the shut state.

Step 14	**exit**
	Example:
	`switch(config-erspan-src)# exit`
	`switch(config)#`
		Exits the monitor configuration mode.

Step 15	(Optional) **show monitor session** {**all** | *session-number* | **range** *session-range*} [**brief**] **Example:** `switch(config)# show monitor session 3`	Displays the ERSPAN session configuration.
Step 16	(Optional) **show running-config monitor** **Example:** `switch(config)# show running-config monitor`	Displays the running ERSPAN configuration.
Step 17	(Optional) **show startup-config monitor** **Example:**	Displays the ERSPAN startup configuration.

	Command or Action	Purpose
	`switch(config)# show startup-config monitor`	
Step 18	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring SPAN Forward Drop Traffic for ERSPAN Source Session

You can configure the device to match on the forwarding drop event and send the matching packets to ERSPAN destination. Doing so can help you to analyze and isolate packet drops in the network.

SUMMARY STEPS

1. **configure terminal**
2. **monitor session** {*session-number* | **all**} **type erspan-source**
3. **vrf** *vrf-name*
4. **destination ip** *ip-address*
5. **source forward-drops rx**
6. **no shut**
7. (Optional) **show monitor session** {**all** | *session-number* | **range** *session-range*}

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**monitor session** {*session-number* | **all**} **type erspan-source**
	Example:
	`switch(config)# monitor session 1 type erspan-source`
	`switch(config-erspan-src)#`
		Configures an ERSPAN source session.

Step 3	**vrf** *vrf-name* **Example:** `switch(config-erspan-src)# vrf default`	Configures the VRF that the ERSPAN source session uses for traffic forwarding.
Step 4	**destination ip** *ip-address* **Example:** `switch(config-erspan-src)# destination ip 10.1.1.1`	Configures the destination IP address in the ERSPAN session. Only one destination IP address is supported per ERSPAN source session.
Step 5	**source forward-drops rx** **Example:**	Configures the SPAN forward drop traffic for the ERSPAN source session.

	Command or Action	Purpose
	`switch(config-erspan-src)# source forward-drops rx`	
Step 6	**no shut** **Example:** `switch(config-erspan-src)# no shut`	Enables the ERSPAN source session. By default, the session is created in the shut state. **Note** Only two ERSPAN source sessions can be running simultaneously.

Step 7	(Optional) **show monitor session** {**all** | *session-number* | **range** *session-range*}
	Example:
	`switch(config-erspan-src)# show monitor session 3`
		Displays the ERSPAN session configuration.

Example

```
switch# config t
  switch(config)# monitor session 1 type erspan-source
  switch(config-erspan-src)# vrf default
  switch(config-erspan-src)# destination ip 40.1.1.1
  switch(config-erspan-src)# source forward-drops rx
  switch(config-erspan-src)# no shut
  switch(config-erspan-src)# show monitor session 1
```

Shutting Down or Activating an ERSPAN Session

You can shut down ERSPAN sessions to discontinue the copying of packets from sources to destinations. You can shut down one session in order to free hardware resources to enable another session. By default, ERSPAN sessions are created in the shut state.

You can enable ERSPAN sessions to activate the copying of packets from sources to destinations. To enable an ERSPAN session that is already enabled but operationally down, you must first shut it down and then enable it. You can shut down and enable the ERSPAN session states with either a global or monitor configuration mode command.

SUMMARY STEPS

1. **configure terminal**
2. **monitor session** {*session-range* | **all**} **shut**
3. **no monitor session** {*session-range* | **all**} **shut**
4. **monitor session** *session-number* **type erspan-source**
5. **shut**
6. **no shut**
7. **exit**
8. (Optional) **show monitor session all**
9. (Optional) **show running-config monitor**
10. (Optional) **show startup-config monitor**
11. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**monitor session** {*session-range* | **all**} **shut** **Example:** `switch(config)# monitor session 3 shut`	Shuts down the specified ERSPAN sessions. By default, sessions are created in the shut state.
Step 3	**no monitor session** {*session-range* | **all**} **shut** **Example:** `switch(config)# no monitor session 3 shut`	Resumes (enables) the specified ERSPAN sessions. By default, sessions are created in the shut state. If a monitor session is enabled but its operational status is down, then to enable the session, you must first specify the **monitor session shut** command followed by the **no monitor session shut** command.

Step 4	**monitor session** *session-number* **type erspan-source**
	Example:
	`switch(config)# monitor session 3 type`
	`erspan-source`
	`switch(config-erspan-src)#`
		Enters the monitor configuration mode for the ERSPAN source type. The new session configuration is added to the existing session configuration.

Step 5	**shut** **Example:** `switch(config-erspan-src)# shut`	Shuts down the ERSPAN session. By default, the session is created in the shut state.
Step 6	**no shut** **Example:** `switch(config-erspan-src)# no shut`	Enables the ERSPAN session. By default, the session is created in the shut state.

Step 7	**exit**
	Example:
	`switch(config-erspan-src)# exit`
	`switch(config)#`
		Exits the monitor configuration mode.

Step 8	(Optional) **show monitor session all** **Example:** `switch(config)# show monitor session all`	Displays the status of ERSPAN sessions.
Step 9	(Optional) **show running-config monitor** **Example:** `switch(config)# show running-config monitor`	Displays the ERSPAN running configuration.
Step 10	(Optional) **show startup-config monitor** **Example:**	Displays the ERSPAN startup configuration.

	Command or Action	Purpose
	`switch(config)# show startup-config monitor`	
Step 11	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring an ERSPAN ACL

You can create an IPv4 or IPv6 ERSPAN ACL on the device and add rules to it.

Before you begin

To modify the DSCP value or the GRE protocol, you need to allocate a new destination monitor session. A maximum of four destination monitor sessions are supported.

SUMMARY STEPS

1. **configure terminal**
2. **{ ip | ipv6 } access-list** *acl-name*
3. [*sequence-number*] {**permit** | **deny**} *protocol source destination* [*protocol-value*]
4. **exit**
5. **vlan access-map** *list-name*
6. **match ip address** *acl-name*
7. **actions (drop | forward | redirect }**
8. **exit**
9. (Optional) **show ip access-lists** *name*
10. (Optional) **show monitor session** {**all** | *session-number* | **range** *session-range*} [**brief**]
11. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**{ ip	ipv6 } access-list** *acl-name*
		Example:
		`switch(config)# ip access-list erspan-acl`
		`switch(config-acl)#`
			Creates the ERSPAN ACL and enters IP ACL configuration mode. The *acl-name* argument can be up to 64 characters.

Step 3	[*sequence-number*] {**permit**	**deny**} *protocol source destination* [*protocol-value*] **Example:**	Creates a rule in the ERSPAN ACL. You can create many rules. The *sequence-number* argument can be a whole number between 1 and 4294967295.

	Command or Action	Purpose

	`switch(config-acl)# permit ip 192.168.2.0/24`
	`switch (config-acl)#`
		The **permit** and **deny** commands support many ways of identifying traffic.

Step 4	**exit**
	Example:
	`switch (config-acl)# exit`
	`switch(config)#`
		Exits the IP ACL configuration mode and enters the global configuration mode.

Step 5	**vlan access-map** *list-name*
	Example:
	`switch(config)# permit ip 192.168.2.0/24`
	`switch(config-access-map)#`
		Creates a VLAN access map and enters the access map configuration mode.

Step 6	**match ip address** *acl-name*
	Example:
	`switch(config-access-map)# match ip address erspan-acl`
	`switch(config-access-map)#`
		Configures the access map to match IP addresses based on the IP ACL configuration.

Step 7	**actions (drop | forward | redirect }**
	Example:
	`switch(config-access-map)# action forward`
	`switch(config-access-map)#`
		Configures the access map to take action on packets whose IP address matches that of the IP ACL configuration.

Step 8	**exit**
	Example:
	`switch (config-access-map)# exit`
	`switch(config)#`
		Exits the access map configuration mode and enters the global configuration mode.

Step 9	(Optional) **show ip access-lists** *name* **Example:** `switch(config)# show ip access-lists erpsan-acl`	Displays the ERSPAN ACL configuration.
Step 10	(Optional) **show monitor session** {**all** | *session-number* | **range** *session-range*} [**brief**] **Example:** `switch(config)# show monitor session 1`	Displays the ERSPAN session configuration.
Step 11	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring UDF-Based ERSPAN

You can configure the device to match on user-defined fields (UDFs) of the outer or inner packet fields (header or payload) and to send the matching packets to the ERSPAN destination. Doing so can help you to analyze and isolate packets that are defined in the criteria by the user.

Before you begin

Make sure that the appropriate TCAM region (SPAN) has been configured using the **hardware access-list tcam region** command to provide enough free space to enable UDF-based ERSPAN. For information, see the "Configuring ACL TCAM Region Sizes" section in the *Cisco Nexus 3400-S Series NX-OS Security Configuration Guide*.

SUMMARY STEPS

1. **configure terminal**
2. **udf** *udf-name offset-base offset length*
3. **hardware access-list tcam region span qualify udf** *udf-names*
4. **copy running-config startup-config**
5. **reload**
6. **ip access-list** *erspan-acl*
7. Enter one of the following commands:
   - **permit udf** *udf-name value mask*
   - **permit ip** *source destination* **udf** *udf-name value mask*
8. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**udf** *udf-name offset-base offset length*
	Example:
	`switch(config)# udf udf-x packet-start 12 1`
	`switch(config)# udf udf-y header outer l3 20 2`
		Defines the UDF as follows:
		• *udf-name*—Specifies the name of the UDF. You can enter up to 16 alphanumeric characters for the name.
		• *offset-base*—Specifies the UDF offset base as follows, where **header** is the packet header to consider for the offset: **packet-start** | **header** {**outer** | **inner** {**l3** | **l4**}}.
		• *offset*—Specifies the number of bytes offset from the offset base. To match the first byte from the offset base (Layer 3/Layer 4 header), configure the offset as 0.
		• *length*—Specifies the number of bytes from the offset. Only 1 or 2 bytes are supported. To match additional bytes, you must define multiple UDFs.

	Command or Action	Purpose
		You can define multiple UDFs, but Cisco recommends defining only required UDFs.
Step 3	**hardware access-list tcam region span qualify udf** *udf-names* **Example:** `switch(config)# hardware access-list tcam region span qualify udf udf-x udf-y`	Attaches the UDFs to one of the following TCAM regions: • **span**—Applies to layer 2 and Layer 3 ports. You can attach up to 2 UDFs to a TCAM region. **Note** Make sure enough free space is available; otherwise, this command will be rejected. If necessary, you can reduce the TCAM space from unused regions and then re-enter this command. For more information, see the "Configuring ACL TCAM Region Sizes" section in the *Cisco Nexus 3400-S Series NX-OS Security Configuration Guide*. **Note** The **no** form of this command detaches the UDFs from the TCAM region and returns the region to single wide.
Step 4	Required: **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.
Step 5	Required: **reload** **Example:** `switch(config)# reload`	Reloads the device. **Note** Your UDF configuration is effective only after you enter **copy running-config startup-config** + **reload**.

Step 6	**ip access-list** *erspan-acl*
	Example:
	`switch(config)# ip access-list erspan-acl-udf-only`
	`switch(config-acl)#`
		Creates an IPv4 access control list (ACL) and enters IP access list configuration mode.

Step 7	Enter one of the following commands:
	• **permit udf** *udf-name value mask*
	• **permit ip** *source destination* **udf** *udf-name value mask*
	Example:
	`switch(config-acl)# permit udf udf-x 0x40 0xF0 udf-y 0x1001 0xF00F`
	Example:
	`switch(config-acl)# permit ip 10.0.0./24 any udf udf-x 0x02 0x0F udf-y 0x1001 0xF00F`
		Configures the ACL to match only on UDFs (example 1) or to match on UDFs along with the current access control entries (ACEs) for the outer packet fields (example 2).
		A single ACL can have ACEs with and without UDFs together. Each ACE can have different UDF fields to match, or all ACEs can match for the same list of UDFs.

	Command or Action	Purpose
Step 8	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuration Examples for ERSPAN

Configuration Example for a Unidirectional ERSPAN Session

This example shows how to configure a unidirectional ERSPAN session:

```
switch# configure terminal
switch(config)# interface ethernet 14/30
switch(config-if)# no shut
switch(config-if)# exit
switch(config)# no monitor session 3
switch(config)# monitor session 3
switch(config-erspan-src)# source interface ethernet 2/1-3 rx
switch(config-erspan-src)# erspan-id 1
switch(config-erspan-src)# ip ttl 16
switch(config-erspan-src)# ip dscp 5
switch(config-erspan-src)# vrf default
switch(config-erspan-src)# destination ip 9.1.1.2
switch(config-erspan-src)# no shut
switch(config-erspan-src)# exit
switch(config)# show monitor session 1
```

Configuration Example for an ERSPAN ACL

The examples in this sectio show how to configure ERSPAN ACLs for both IPv4 and IPv6.

This example shows how to configure an ERSPAN IPv4 ACL:

```
switch# configure terminal
switch(config)# ip access-list match_11_pkts
switch(config-acl)# permit ip 11.0.0.0 0.255.255.255 any
switch(config-acl)# exit
switch(config)# ip access-list match_12_pkts
switch(config-acl)# permit ip 12.0.0.0 0.255.255.255 any
switch(config-acl)# exit
switch(config)# vlan access-map erspan_filter 5
switch(config-access-map)# match ip address match_11_pkts
switch(config-access-map)# action forward
switch(config-access-map)# exit
switch(config)# vlan access-map erspan_filter 10
switch(config-access-map)# match ip address match_12_pkts
switch(config-access-map)# action forward
switch(config-access-map)# exit
switch(config)# monitor session 1 type erspan-source
switch(config-erspan-src)# filter access_group erspan_filter
```

This example shows how to configure an ERSPAN IPv6 ACL:

```
switch# configure terminal
switch(config)# ipv6 access-list match_11_pkts
switch(config-acl)# permit ipv6 permit ipv6 2040::0/32 any
switch(config-acl)# exit
switch(config)# vlan access-map erspan_filter 15
switch(config-access-map)# match ipv6 address match_11_pkts
switch(config-access-map)# action forward
switch(config-access-map)# exit
switch(config)# monitor session 1 type erspan-source
switch(config-erspan-src)# filter access_group erspan_filter
```

Configuration Examples for UDF-Based ERSPAN

This example shows how to configure UDF-based ERSPAN to match on the inner TCP flags of an encapsulated IP-in-IP packet using the following match criteria:

- Outer source IP address: 10.0.0.2
- Inner TCP flags: Urgent TCP flag is set
- Bytes: Eth Hdr (14) + Outer IP (20) + Inner IP (20) + Inner TCP (20, but TCP flags at 13th byte)
- Offset from packet-start: 14 + 20 + 20 + 13 = 67
- UDF match value: 0x20
- UDF mask: 0xFF

```
udf udf_tcpflags packet-start 67 1
hardware access-list tcam region span qualify udf udf_tcpflags
copy running-config startup-config
reload
ip access-list acl-udf
  permit ip 10.0.0.2/32 any udf udf_tcpflags 0x20 0xff
monitor session 1 type erspan-source
  source interface Ethernet 1/1
  filter access-group acl-udf
```

This example shows how to configure UDF-based ERSPAN to match regular IP packets with a packet signature (DEADBEEF) at 6 bytes after a Layer 4 header start using the following match criteria:

- Outer source IP address: 10.0.0.2
- Inner TCP flags: Urgent TCP flag is set
- Bytes: Eth Hdr (14) + IP (20) + TCP (20) + Payload: 112233445566DEADBEEF7788
- Offset from Layer 4 header start: 20 + 6 = 26
- UDF match value: 0xDEADBEEF (split into two-byte chunks and two UDFs)
- UDF mask: 0xFFFFFFFF

```
udf udf_pktsig_msb header outer l3 26 2
udf udf_pktsig_lsb header outer l3 28 2
hardware access-list tcam region span qualify udf udf_pktsig_msb udf_pktsig_lsb
copy running-config startup-config
reload
ip access-list acl-udf-pktsig
  permit udf udf_pktsig_msb 0xDEAD 0xFFFF udf udf_pktsig_lsb 0xBEEF 0xFFFF
monitor session 1 type erspan-source
  source interface Ethernet 1/1
  filter access-group acl-udf-pktsig
```