

L7 PCC Rules

- Revision History, on page 1
- Feature Description, on page 1
- How It Works, on page 2

Revision History

Note Revision history details are not provided for features introduced before release 21.24.

Revision Details	Release
First introduced	Pre 21.24

Feature Description

With this feature, the L7 analyzer functionality is supported in the CUPS architecture.

The following L7 analyzers are supported:

- HTTP
- HTTPS
- RTP/RTSP
- FTP
- DNS
- Content Filtering
- DNS Snooping

The following charging actions are supported:

- Discard

- Terminate Flow
- Redirect (if applicable)

How It Works

This section provides a brief overview of the L7 analyzer functionality that are supported as part of this feature.

Content Filtering

Content Filtering is an in-line service available for 3GPP and 3GPP2 networks to filter HTTP requests from mobile subscribers based on the URLs in the requests. This enables operators to filter and control the content that an individual subscriber can access, so that subscribers are inadvertently not exposed to universally unacceptable content and/or content inappropriate as per the subscribers' preferences.

The Content Filtering functionality remains the same as implemented in the non-CUPS architecture. For more information, refer to *Content Filtering Support Overview* chapter in the *CF Administration Guide*.

Configuring the Content Filtering

Use the following additional configuration to enable the content filtering:

```
configure
   require user-plane content-filtering
      content-filtering category database directory path path_address
      content-filtering category database max-version version_number
      end
```

Note The above configuration must be configured on the User Plane, during boot time, to enable Content Filtering. Defining the above configuration post the User Plane configuration will lead to errors and inconsistencies.

Note To enable the feature, license for User Plane as well as existing content filtering license is required on Uplane.

Note For ICSR User Plane 1:1, the database is loaded pn both the UP's, separately. The rest of the Content Filtering configurations on Control Plane remains the same. The Content Filtering configuration is pushed from Control Plane to active the User Plane and then to standby User Plane.

Configuration on Control Plane

The following sample configuration demonstrates changes required on Control Plane for Content Filtering:

```
config
            active-charging-service ACS
               content-filtering category policy-id 1
               analyze priority 1 category ABOR
```

```
              analyze priority 2 category ADVERT action allow
             analyze priority 2 category ADVERT action allow action content-insert "Content
Restricted : The Web Guard feature has been enabled on your line. Web Guard has restricted
your access to this content. The person on your Wireless account who is designated as the
Primary Account Holder can disable this restriction through the account management website"

              exit
           rulebase cisco
              content-filtering mode category static-only
              content-filtering flow-any-error permit
              content-filtering category policy-id 5
```

The configuration on the Control Plane is pushed to User Plane using the PFD mechanism.

Use the following show commands to validate the content filtering configuration on User Plane:

- show user-plane-service rulebase name cisco
- show user-plane-service content-filtering category policy-id

Use the following show commands to check the CFDB spawning on User Plane:

- show content-filtering category database facility srdbmgr
- show content-filtering category database verbose debug-only
- show content-filtering category database verbose
- show content-filtering category database url
- show content-filtering category url

The Content Filtering policy ID received from PCRF for a particular subscriber is sent to User Plane during call establishment. The PFCP messages Sx establishment request/Sx modify request contains the CF Policy ID.

Use the following command to check the CF Policy Id on User Plane:

show subscribers user-plane-only callid full all

The following filed is displayed in support of Content Filtering in CUPS:

- Content Filtering Policy ID

Use the following show commands to monitor the SRDB Request/Response/CF Polict actions:

- show user-plane-service inline-services content-filtering category statistics
- show user-plane-service inline-services content-filtering category statistics rulebase name
- show content-filtering category statistics
- show content-filtering category statistics facility srdbmgr instance 1
- show content-filtering category statistics volume all



Note All existing bulk statistics defined for Content Filtering in the non-CUPS architecture is also applicable in CUPS.

Limitations

- Dynamic content filtering mode is not supported.
- Rulebase command **content-filtering flow-any-error [permit | deny]** is not supported.

DNS

Offloading to SM-P

DNS packets are not offloaded to SM-P.

Charging

DNS packets are charged at SM-P.

Rule Matching

The functionality remains the same as the non-CUPS architecture.

Statistics

Use the following CLI command to get statistics related to DNS: **show user-plane-service statistics analyzer name dns**

DNS Snooping

Charging

The charging of DNS Snooping takes place at SM-P.

Rule Definitions

Use the following CLI commands for specifying the rule definition hostnames (domain-names) and part of the host names.

```
ruledef <ruledef_name>
      ip [server-domain-name {contains|=|ends-with|starts-with} <url_string>]

      ip [server-domain-name {contains|=|ends-with|starts-with} <url_string>]

       multi-line-OR enabled
```

Use the no version of this CLI to delete the ruleline for ip server- domain-name.

```
ruledef <ruledef_name>
  no ip [server-domain-name {contains|=|ends-with|starts-with} <url_string>]

   exit
```

Use the following CLI for configurable timer of DNS entries at ECS level.

```
configure
        active-charging service service_name
```

```
ip dns-resolved-entries timeout <value_secs>
end
```

Whenever the ruledef containing the ip server-domain-name keyword is defined and used in rulebase, the ip-table is created per rulebase per instance.

Rule Matching

The functionality remains the same as the non-CUPS architecture.

Show CLIs

Use the following CLIs to check the table for DNS IP entries:**show user-plane-service [statistics dns-learnt-ip-addresses {summary | sessmgr instance <id> |all [verbose] }]**

Bulkstats

The following bulkstats are available in support of DNS Snooping feature:

- ecs-dns-learnt-ipv4-entries
- ecs-dns-flushed-ipv4-entries
- ecs-dns-replaced-ipv4-entries
- ecs-dns-overflown-ipv4-entries
- ecs-dns-learnt-ipv6-entries
- ecs-dns-flushed-ipv6-entries
- ecs-dns-replaced-ipv6-entries
- ecs-dns-overflown-ipv6-entries

The above bulkstats are added in the ECS schema same as in the non-CUPS architecture.



Note The SNMP Trap generation commands are not supported in CUPS DNS snooping feature.

FTP

Offloading to SM-P

Only for FTP data, TRM is engaged. FTP data flows are eligible for offloading to SM-P.

There is no TRM engagement for control FTP flows.

Charging

FTP packets are charged at SM-P.

Rule Matching

The functionality remains the same as the non-CUPS architecture.

Statistics

Use the following CLI command to get statistics related to FTP: **show user-plane-service statistics analyzer name ftp**

HTTP

HTTP Offloading to SM-P

On a header completion of HTTP Request/Response, the uplink/downlink data packets are offloaded to VPP in the following cases:

- Content-Length – Volume-based offloading is supported for methods like GET and POST. The HTTP flow with chunk-encoding data transfer mechanism does not get offloaded irrespective of the method defined in HTTP. If the stream is offloaded based on content-length, then the stream on the other end will also get offloaded until the former is not onloaded.
- CONNECT Method—The method where both uplink and downlink streams are offloaded after flow is upgraded to CONNECT.
- WebSocket Method—After the flow is classified as WebSocket protocol, both uplink and downlink streams are offloaded.
- The streams are onloaded back to SM-U application in either of the following cases:
  - FIN packet received.
  - Content-length is breached.
  - PDN update.

Header Parsing

Similar to non-CUPS implementation, only the header fields defined in ruledefs, which are included in rulebase, are parsed. Or, in case of features like x-header, redirection is configured which have dependencies on some of the HTTP header fields.

Rule Matching

There is no functional change in the way rule-matching takes place in CUPS. The only change is specific to TRM wherein both uplink and downlink has its own TRM.

HTTP Charging

- Complete Packets are charged at SM-P.
- Partial Packets are charged on SM-U on completion. Packet completing the Partial Packet is also charged on SM-U.
- Concatenated Packets are charged on SM-U.

- Delay Charging is enabled – In case there are uncharged bytes, the packet along with the uncharged bytes gets charged on SM-U.
- Response-based charging is enabled – On receiving a Response, both uplink and downlink packets are charged on SM-U. Subsequent uplink and downlink packets are charged at SM-P, unless they are partial/concatenated.

X-Header Parsing and Rule-Matching

Ruledefs with x-header rule-lines are parsed and matched.

WebSocket

The functionality remains the same as non-CUPS architecture.

TRM and Response-Based Charging

Transactional Rule Matching will only avoid per-packet rule matching after a flow is fully classified.

Direction-based TRM has been introduced in CUPS, wherein there are two TRMs for a flow, one for uplink and the other for downlink direction. After a packet enables TRM, subsequent packets (TRM eligible) continue to match the same rule resulting in efficient rule-matching. That is, uplink packets match the uplink TRM cached rule, and downlink packets match the downlink TRM cached rule.

URL-Based Redirection

The functionality remains the same as non-CUPS architecture.

For flow action redirect-url, encrypt is not supported. Currently, the following dynamic fields are supported:

- #HTTP.URI#
- #HTTP.HOST#
- #HTTP.URL#
- #ACSMGR_BEARER_CALLED_STATION_ID#
- #RULEBASE#
- #RTSP.URI#

X-Header Insertion

X-header Insertion is supported in HTTP Requests. The behavior remains same as that of non-CUPS architecture. With respect to offloading to SM-P:

- Flows, for which X-header is inserted in a packet, are not offloaded.
- With X-header configuration, all TCP OOO packets irrespective of transmit order CLI, will be buffered and sent out after reordering.

X-Header insertion statistics CLI

show user-plane-service statistics charging-action name *charging_action_name*

The following fields are added in support of X-header insertion:

- For Request:
  - XHeader Bytes Injected
  - XHeader Pkts Injected
  - XHeader Bytes Removed
  - XHeader Pkts Removed
  - IP Frags consumed by XHeader

Limitation

- X-Header Spoofing is not supported.
- X-Header Insertion in Response packet is not supported.
- X-Header Encryption with RSA and RC4MD5 is supported but not supported with AES.
- Monitor protocol for X-Header is not supported.
- Following X-Header fields insertion is not supported in a packet: QoS, UIDH, Customer ID, Hash Value, Time of the Day, Radius String, Session-Id, Congestion Level, User-Profile.

HTTP Analyzer Statistics

Use the following CLI command to get statistics related to the HTTP analyzer: **show user-plane-service statistics analyzer name http**

HTTPS

HTTPS Offloading to SM-P

HTTPS flows are offloaded to SM-P after receiving the application packet. With the P2P analyzer, offloading works when P2P analyzer detects the L7 protocol.

HTTPS Charging

Charging for HTTPS packets are done at SM-P.

Statistics

Use the following CLI command to get statistics related to HTTPS: **show user-plane-service statistics analyzer name secure-http**

HTTP URL Filtering

The HTTP URL Filtering feature simplifies rule definitions used for URL detection.

The HTTP request packet can have a proxy (prefixed) URL and an actual URL. If a proxy URL is found in the HTTP request packet, the HTTP URL Filtering feature truncates this URL from the parsed information and only the actual URL is used for rule matching and Event Data Records (EDR) generation.

Configuring the HTTP URL Filtering Feature

This section describes how to configure the HTTP URL Filtering feature.

Configuring Group of Prefixed URLs

To configure the group of prefixed URLs, use the following CLI commands:

```
configure
  active-charging service ecs_service_name
    group-of-prefixed-urls prefixed_urls_group_name
    end
```

Configuring URLs in the Group of Prefixed URLs

To configure URLs to be filtered in the group of prefixed URLs, use the following CLI commands:

```
configure
  active-charging service ecs_service_name
    group-of-prefixed-urls prefixed_urls_group_name
      prefixed-url url_1
      ...
      prefixed-url url_10
      end
```

Enabling the Group of Prefixed URLs in Rulebase

To enable the group of prefixed URLs in rulebase for processing prefixed URLs, use the following CLI commands:

```
configure
  active-charging service ecs_service_name
    rulebase rulebase_name
      url-preprocessing bypass group-of-prefixed-urls
prefixed_urls_group_name_1
      ...
      url-preprocessing bypass group-of-prefixed-urls
prefixed_urls_group_name_64
      end
```

This configuration on the control plane chassis will be pushed to the user plane with a PFD message for “group-of-prefixed-urls” and “rulebase-url-preprocessing”separately.

The group of prefixed URLs has the list of proxy URLs, which must be truncated. The rulebase contains multiple group of prefixed urls, which must be filtered. Charging ruledefs contain rules for actual URLs that must be searched after truncating URLs in the group of prefixed URLs.



Note

- Each group of prefixed URLs can have a maximum of ten prefixed URLs.
- A maximum of 64 group of prefixed URLs can be created and configured.

Show Commands

show user-plane-service group-of-prefixed-urls all | name *group_name*

This show command can be used on the user plane to verify whether the group of prefixed URLs are pushed or not. The output of this command is as follows:

- Name of the group of prefixed URLs
- Prefixed URLs
- Total number of prefixed URLs found

show user-plane-service rulebase name *rbase_name*

This show command can be used on the user plane to check whether the group of prefixed URLs is configured in rulebase or not. The output of this command is as follows:

- Name of rulebase
- Name of the groups of prefixed Urls for URL pre-processing

show user-plane-service statistics analyzer name http

The output of this command is as follows:

- Total HTTP Sessions
- Current HTTP Sessions
- Total Uplink Bytes
- Total Downlink Bytes
- Total Uplink Pkts
- Total Downlink Pkts
- Uplink Bytes Retrans
- Downlink Bytes Retrans
- Uplink Pkts Retrans
- Downlink Pkts Retrans
- Total Request Succeed
- Total Request Failed
- GET Requests
- POST Requests
- CONNECT Requests
- PUT requests
- HEAD requests
- Websocket Flows
- Invalid packets

- Wrong FSM packets
- Unknown request method
- Pipeline overflow requests
- Corrupt request packets
- Corrupt response packets
- Unhandled request packets
- Unhandled response packets
- Partial HTTP Header Anomaly prevented
- New requests on closed connection
- Memory allocation failures
- Packets after permanent failure
- Prefixed Urls Bypassed
- FastPath Statistics
- Total FP Flows
- Uplink (Total FP Pkts)
- Downlink (Total FP Pkts)
- Uplink (Total FP Bytes)
- Downlink (Total FP Bytes)



Note Prefixed URLs Bypassed counter has been added in http analyzer stats as a performance measurement to show the number of truncated prefixed URLs.

RTP/RTSP

Offloading to SM-P

RTP, being on UDP Protocol, is offloaded immediately.

RTSP flow is not offloaded. There is no TRM engagement for RTSP flows.

Charging

RTP packets are charged at SM-P. RTSP packets are charged at SM-P unless the packets being partial or if delay-charging is enabled.

Rule Matching

The functionality remains the same as the non-CUPS architecture.

Statistics

Use the following CLI command to get statistics related to RTP: **show user-plane-service statistics analyzer name rtp**

Use the following CLI commands to get statistics related to RTSP:

- **show user-plane-service statistics analyzer name rtsp**
- **show user-plane-service statistics analyzer name rtsp verbose**

RTP Dynamic Flow Detection

The **rtp dynamic-flow-detection** CLI command, under the ACS Rulebase Configuration mode, enables the Real Time Streaming Protocol (RTSP) and Session Description Protocol (SDP) analyzers to detect the child RTP and RTCP flows. If you configure the RTSP/SIP and SDP analyzers, and **rtp dynamic-flow-detection** CLI is present, then there's no need for configuring RTP/RTCP explicitly. With the **rtp dynamic-flow-detection** CLI command, the child RTP or RTCP flows get corelated to their parent RTSP/SIP-SDP flows.

Once the parent flow (RTSP/SIP-SDP) gets cleared, the child RTP/RTCP flows also gets cleared. In the absence of this CLI, the L7 layer analysis for RTP and RTCP needs a separate analyzer configuration. There’s no correlation of RTP/RTCP flows to RTSP/SIP-SDP flow.

Rule-matching for Bearer-specific Filters

Rule Matching

The functionality remains the same as the non-CUPS architecture.

IMSI-based rules are matched as per the subscribers IMSI.

APN-based rules allows you to define rule expressions to match Access Point Name (APN) of the bearer flow.

RAT-Type allows you to define rule expressions to match Radio Access Technology (RAT) in the bearer flow.

Rule Definitions

Use the following CLI commands to configure the IMSI pool.

```
configure
  active-charging service service_name
    imsi-pool pool_name
      imsi { imsi_number | range start_imsi to end_imsi }
```

The imsi-pool can contain either IMSI value or range of IMSI.

Use the following CLI commands to configure rule line under ruledef.

```
configure
  active-charging service service_name
    ruledef ruledef_name
      bearer 3gpp imsi { = imsi_value } | { range imsi-pool pool_name }
      bearer 3gpp apn operator apn_name
      bearer 3gpp rat-type operator rat_type
```

IMSI range can be configured in a rule with the help of IMSI pool.

For more information about the CLI commands, see *ACS Ruledef Configuration Mode Commands* in the StarOS *Command Line Interface Reference*.

Show CLIs

Use the following CLI on User Plane to see information about IMSI pool that is configured in a service:**show user-plane-service imsipool name** *pool_name*

SIP

Offloading to SM-P

SIP flow is not offloaded.

Charging

SIP packets are charged at SM-P.

Rule Matching

The functionality remains the same as the non-CUPS architecture.

Statistics

Use the following CLI command to get statistics related to SIP: **show user-plane-service statistics analyzer name sip**