ERICSSON

PISC Configuration

OPERATION DIRECTIONS

56/1543-AXB 250 10/4-V3 Uen JC

Contents

1 Introduction

This document describes the configuration of the Packet Inspection and Service Classification (PISC) feature in the Service-Aware Charging and Control (SACC) business solution on the EPG product. The configuration is covered for both the GGSN and PGW functionality.

1.1 Scope

This document covers the following issues:

— Enabling PISC, with mandatory configuration steps to achieve basic PISC functionality

— Configuring rules for header packet inspection

— Configuring Deep Packet Inspection (DPI)

— Configuring heuristic packet inspection

It is assumed that all the required licenses are set up correctly for the features described in this document. For more information about licensed features, refer to EPG Features. For more information on how to configure licenses, refer to Software License Management and Licensing.

For an overview and technical description of PISC, refer to Packet Inspection and Service Classification (PISC).

For detailed information and recommendations that help optimize the SACC-related configuration, refer to SACC Optimization.

Attention!

After the configuration change, all state-related information is lost and all traffic is analyzed again without optimization. Therefore, a configuration change temporarily decreases the throughput and increases the latency, CPU load, and memory. During a PISC configuration change, bearer creation and user packets can be affected.

1.2 Target Groups

This document is intended for personnel performing PISC-related configuration in the GGSN or PGW within the EPG product. The document assumes a basic knowledge of data communication and telecommunication.

2 Enable PISC

To enable PISC for a rule space, do the following:

1. Activate the PISC feature, as described in Section 2.1 on page 2.
2. Configure a service set, as described in SACC Configuration.
3. Configure a default SDF for the service set, as described in SACC Configuration.
4. Configure header packet inspection for the service set, as described in Section 3 on page 2.
5. Optionally, configure DPI for the service set, as described in Section 4 on page 14.
6. Optionally, configure heuristic packet inspection for the service set, as described in Section 5 on page 72.
7. Associate the service set with the rule space, as described in SACC Configuration.
8. Optionally, configure the maximum number of DPI flows per protocol and user, per node, and the maximum accepted dynamic routing rules per protocol, as described in Section 6.2 on page 78.

2.1 Activate the PISC Feature

To activate the PISC feature, include the following statement:

```
Ericsson(config)# epg node feature-activation
   packet-inspection
```

3 Configure Header Packet Inspection

To configure header packet inspection, the following actions are mandatory:

— Configure at least one header rule. For further information, see Section 3.1 on page 2.

— Associate the header rule with a header rule set. For further information, see Section 3.2 on page 11.

— Associate the header rule set with a service set. For further information, see Section 3.3 on page 11.

3.1 Configure Header Rule

A header rule consists of one or several terms. The terms are evaluated in descending order by a user-defined `term-id`. The `term-id` is a numerical value from 1 to 999,999. The EPG capacity is only affected by the total number of terms, refer to EPG Characteristics.

To configure a term in a header rule, take the following actions:

— Configure the match conditions. If several match conditions are configured in a term, all conditions must be fulfilled for the term to match. Multiple match conditions in the same term are handled in a logical `AND` relation.

— Configure the unique resulting SDF-ID.

3.1.1 Configure Match Conditions

The following match conditions can be configured for a term in a header rule:

— UE prefix

— UE address

— UE port

— Network prefix

— Network address

— Network port

— Protocol

— Traffic type

— Domain

Incoming packets are identified by UE IP address (expressed as UE prefix or UE address), UE communication port, network IP address (expressed as network prefix or network address), network communication port, protocol, traffic type, and domain. Header rule match conditions are used to filter incoming packets depending on one or more of these attributes.

When configuring match conditions for multiple rules for different protocols, ensure that no incoming packet satisfies two or more header rule match conditions for different protocols at the same time. Incoming packets that satisfy multiple header rule match conditions for different protocols can cause classification errors.

The following sections describe how to configure each of these match conditions.

3.1.1.1 UE Prefix

To configure packet matching based on an IPv4 or IPv6 UE prefix, include the following statement:

```
Ericsson(config)# epg pgw service-identification header-rule
<rule-name> term <term-id> from
   ms-prefix [<prefix>]
```

3.1.1.2 UE Address

To configure packet matching based on a complete IPv4 or IPv6 UE address, include the following statement:

```
Ericsson(config)# epg pgw service-identification header-rule
<rule-name> term <term-id> from
   ms-address [<address>]
```

3.1.1.3 UE Port

To configure packet matching based on the UE port, include the following statement:

```
Ericsson(config)# epg pgw service-identification header-rule
<rule-name> term <term-id> from
   ms-port <port-number>
```

The port can be defined either with its numerical value or with one of the pre-defined keywords. A range of ports can also be specified.

The following pre-defined keywords are supported for the `ms-port` configuration: `afs`, `bgp`, `biff`, `bootpc`, `bootps`, `cmd`, `cvspserver`, `dhcp`, `domain`, `eklogin`, `ekshell`, `exec`, `finger`, `ftp`, `ftp-data`, `ftpes`, `ftps`, `ftps-data`, `http`, `https`, `ident`, `imap`, `imaps`, `kerberos-sec`, `klogin`, `kpasswd`, `krb-prop`, `krbupdate`, `kshell`, `ldap`, `ldp`, `login`, `mobileip-agent`, `mobilip-mn`, `msdp`, `netbios-dgm`, `netbios-ns`, `netbios-ssn`, `nfsd`, `nntp`, `ntalk`, `ntp`, `pop3`, `pops`, `pptp`, `printer`, `radacct`, `radius`, `rip`, `rkinit`, `smtp`, `smtps`, `snmp`, `snmptrap`, `snpp`, `socks`, `ssh`, `sunrpc`, `syslog`, `tacacs`, `tacacs-ds`, `talk`, `telnet`, `tftp`, `timed`, `wap-wsp`, `wap-wtp-wsp`, `who`, `xdmcp`

The `wap-wsp` option matches port `9200` and the `wap-wtp-wsp` option matches port `9201`.

3.1.1.4 Network Prefix

To configure packet matching based on an IPv4 or IPv6 network prefix, include the following statement:

```
Ericsson(config)# epg pgw service-identification header-rule
<rule-name> term <term-id> from
```

```
   network-prefix [<prefix>]
```

The network-prefix is an IPv4 or IPv6 address in canonical format followed by a mask value. The maximum mask value is 32 for IPv4 and 128 for IPv6.

3.1.1.5 Network Address

To configure packet matching based on a complete IPv4 or IPv6 network address, include the following statement:

```
Ericsson(config)# epg pgw service-identification header-rule
<rule-name> term <term-id> from
   network-address [<address>]
```

3.1.1.6 Network Port

To configure packet matching based on the network port, include the following statement:

```
Ericsson(config)# epg pgw service-identification header-rule
<rule-name> term <term-id> from
   network-port <port-number>
```

The port can be defined either with its numerical value or with one of the pre-defined keywords. A range of ports can also be specified.

The following pre-defined keywords are supported for the network-port configuration: afs, bgp, biff, bootpc, bootps, cmd, cvspserver, dhcp, domain, eklogin, ekshell, exec, finger, ftp, ftp-data, ftpes, ftps, ftps-data, http, https, ident, imap, imaps, kerberos-sec, klogin, kpasswd, krb-prop, krbupdate, kshell, ldap, ldp, login, mobileip-agent, mobilip-mn, msdp, netbios-dgm, netbios-ns, netbios-ssn, nfsd, nntp, ntalk, ntp, pop3, pops, pptp, printer, radacct, radius, rip, rkinit, smtp, smtps, snmp, snmptrap, snpp, socks, ssh, sunrpc, syslog, tacacs, tacacs-ds, talk, telnet, tftp, timed, wap-wsp, wap-wtp-wsp, who, xdmcp

The wap-wsp option matches port 9200, the wap-wtp-wsp option matches port 9201.

The following apply to a header rule with an associated HTTP-WSP set:

— Network port can only be specified if the transport protocol, User Datagram Protocol (UDP) or Transmission Control Protocol (TCP), is specified.

— If no protocol or network port is configured in the header rule, this rule matches Hypertext Transfer Protocol (HTTP), Wireless Session Protocol (WSP), and connectionless WSP traffic to the default ports (HTTP: 80/8080, WSP: 9201, Connectionless WSP: 9200).

— If matching on the TCP is configured and no ports are specified, default HTTP ports (80/8080) are used. This rule does not match WSP or connectionless WSP traffic.

— If matching on the UDP is configured and no port is specified, default ports for WSP (9201) and connectionless WSP (9200) are used. This rule does not match HTTP traffic.

— If matching on the UDP is configured, and one or several ports are specified, port 9200 matches connectionless WSP traffic while all other specified ports are used for connection-oriented WSP traffic.

— If matching on the UDP and TCP is configured simultaneously, it is recommended not to configure specific ports, as the default TCP ports (80, 8080) and UDP ports (9200, 9201) are used to inspect HTTP and WSP traffic.

— If matching on the UDP and TCP is configured simultaneously and other ports than the standard ones are required to inspect HTTP and WSP traffic, separate it into two different terms: the TCP-related ports and the UDP-related ports. Each term points to the corresponding HTTP-WSP rule set, but TCP traffic only evaluates HTTP-related filters and UDP traffic only evaluates WSP-related filters.

```
epg pgw service-identification header-rule all_tcp
 term 1
  name example1
  then service-data-flow-id payload 80
  then protocol-inspection http-wsp-rule-set wapHttpWspRs
  from network-port [ 80 8080 8081 ]
  from protocol tcp
 !
!
```

Example 1 Network Port Configuration for TCP

```
epg pgw service-identification header-rule all_udp
 term 2
  name example2
  then service-data-flow-id payload 80
  then protocol-inspection http-wsp-rule-set wapHttpWspRs
  from network-port [ 9200 9201 9202 9203 ]
  from protocol udp
 !
!
```

Example 2 Network Port Configuration for UDP

3.1.1.7 Protocol

To configure packet matching based on the used protocol, include the following statement:

```
Ericsson(config)# epg pgw service-identification header-rule
<rule-name> term <term-id> from protocol
   ah
   egp
   esp
   gre
   icmp
   icmpv6
   igmp
   ipip
   <number>
   ospf
   pim
   rsvp
   sctp
   tcp
   udp
```

Note: The protocol can only be configured to the values `icmp`, `icmpv6`, `udp`, and `tcp` when the protocol match condition is combined with the following conditions in a header rule:

- An IP address or port, or a combination of both
- Traffic type unsolicited

The transport protocol can be specified as a numeric value from `0` through 255, according to the Internet Assigned Numbers Authority (IANA) listing found here.

3.1.1.8 Traffic Type

To configure packet matching based on unsolicited traffic, include the following statement:

```
Ericsson(config)# epg pgw service-identification header-rule
<rule-name> term <term-id> from
   traffic-type unsolicited
```

The unsolicited traffic type match condition can be configured alone or with UE prefix, UE address, UE port, network prefix, network address, network port, and protocol. When the protocol is not specified, TCP, UDP, ICMP, and ICMPv6 are considered.

3.1.1.9 Domain

To configure packet matching based on the domain, include the following statement:

```
Ericsson(config)# epg pgw service-identification header-rule
<rule-name> term <term-id> from domain
   is <string>
```

```
starts-with <string>
contains <string>
ends-with <string>
```

Note: The maximum number of match conditions that can be configured is 10.

When the match condition is set to domain, the corresponding header rules are referred to as domain-based header rules. When domain-based header rules are configured, DNS monitoring is enabled. See Section 3.1.2 on page 9 for how to configure DNS monitoring.

Note: Only inspection of DNS traffic over UDP is supported.

The following applies to domain-based header rules:

— Only the listed domain conditions such as `is`, `starts-with`, `contains`, `ends-with` are allowed; and only one condition is allowed in a term.

— Only one domain can be specified in a term.

— The domain is not case-sensitive.

— The domain condition can be configured alone, with the protocol condition, or with the protocol and network port conditions. It is highly recommended to configure the additional protocol condition in a term in a domain-based header rule. In such a combination, the protocol condition is limited to the following values: TCP, UDP, TCP and UDP.

— A domain-based header rule can be associated with an SDF-ID, a single DPI rule set, or both. Only the following DPI rule sets can be associated with a domain-based header rule:

  - HTTP-WSP rule set
  - TLS rule set
  - SIP rule set

— HTTP analysis associated with domain-based header rules behaves in the same way as that associated with non-domain-based header rules. When protocol inspection is configured to associate a rule set of HTTP rules with a domain-based header rule, only the default ports 80/8080 are used for HTTP analysis. To perform the traffic analysis for different ports, the network port condition must be specified in the domain-based header rule.

— When configuring domain-based header rules with TCP deferred charging or signalling classification in the same service set, signalling rules (even those based on network address in a header rule) have precedence over other header rules including domain-based header rules. For more information, refer to Packet Inspection and Service Classification (PISC).

Any change in the PISC configuration clears the list of IP addresses discovered through the domain-based header rules and the corresponding dynamic header rules.

3.1.2 Configure DNS Monitoring for Domain-Based Header Rules

This section describes how to configure DNS monitoring for domain-based header rules.

3.1.2.1 Configure Trusted DNS Servers

A list of trusted DNS servers can be defined for domain-based header rules. When such a list is defined, only IPv4 or IPv6 addresses resolved by the trusted DNS servers are considered for domain-based header rules, and no dynamic header rules are created for DNS queries addressed to the DNS servers not included in the list.

When no list of trusted DNS servers is defined, domain-based header rules obtain IPv4 or IPv6 addresses from DNS responses from any DNS server.

To configure a list of trusted DNS servers for domain-based header rules, include the following statement:

```
Ericsson(config)# epg pgw service-identification dns-monitoring trusted-dns-
   network-address <address>
```

3.1.2.2 Clear Dynamic Header Rules and IP Addresses

The list of IP addresses discovered through domain-based header rules, the corresponding dynamic header rules, and DNS cached entries need to be cleared. The clearing is performed either automatically with an expiry mechanism or manually by the operator. The deletion of the dynamic header rules affects new and existing traffic flows if they are not already classified. Existing traffic flows that are already classified are not affected. After the clearing, the affected traffic flows are classified according to other static configuration rules or the default SDF-ID, unless they are preceded by new DNS queries matching the internet side IP addresses of the flows.

The expiry mechanism is based on the Time to Live (TTL) value in the received `DNS Query Response` and a configurable minimum TTL value. The `DNS Query Response` sent from the DNS server carries the time stamp showing when the `DNS Query` was resolved and the TTL value. To avoid low TTL values, a new TTL is set to whichever is greater between the received TTL and the configurable minimum TTL. After receiving the `DNS Query Response`, the EPG calculates the expiry time by adding the new TTL to the time stamp and stores the value.

To configure the minimum TTL, include the following statement:

```
Ericsson(config)# epg pgw service-identification dns-monitoring
   minimum-ttl <seconds>
```

By default, the minimum TTL is set to 10 seconds.

Manual clearing of dynamic header rules and IP addresses is one of the EPG monitoring activities. For more information, refer to Action Commands for the GGSN and PGW.

3.1.2.3 Configure Limit of Dynamic Header Rules

A maximum limit for the number of dynamic header rules needs to be set to avoid processing and memory exhaustion when too many IP addresses are received for matching DNS queries because of attacks or faulty nodes. When the limit is reached, no more dynamic header rules are created because of domain-based header rules and an event with the level of WARNING is logged. After solving the problem, it is recommended to perform a manual clearing of all dynamic header rules generated by domain-based header rules to clear useless rules and allow new valid rules to be created.

To configure the maximum number of generated dynamic header rules, include the following statement:

```
Ericsson(config)# epg pgw service-identification dns-monitoring
   max-shallow-rules <max-rules>
```

The value range of `max-shallow-rules` is 256–512,000. The default value is 25,000.

3.1.2.4 List Dynamic Header Rules

Listing dynamic header rules is one of the EPG monitoring activities. Refer to Action Commands for the GGSN and PGW for more information.

3.1.3 Activate URI Tracking for HTTP Default Classification per Header Rule

To activate URI tracking for HTTP packets that match a header rule fallback SDF-ID, include the following statement:

```
Ericsson(config)# epg pgw service-identification header-rule
<rule-name> term <term-id> then
   http-enable-implicit-uri-tracking
```

Note:

- Activating URI tracking for HTTP default classification in a header rule is not permitted if the header rule fallback SDF-ID is not configured.
- URI tracking for HTTP default classification uses a high percentage of node resources.

3.1.4 Configure Resulting SDF-ID

To configure the SDF-ID to be assigned to traffic that matches the conditions in a term, include the following statement:

```
Ericsson(config)# epg pgw service-identification header-rule
<rule-name> term <term-id> then service-data-flow-id
   payload <sdf-id>
```

Note: Ensure that either the resulting SDF-ID, or the protocol inspection clause is included in each term. Otherwise, the rule is silently discarded, and the traffic can be classified in other rules or in the default SDF-ID.

Example 3 shows an example of configuring a fallback SDF-ID in a header rule term using SIP protocol inspection. For more information on fallback SDF-IDs, refer to SACC Optimization.

```
epg pgw service-identification header-rule hdr-rs1-r1
 term 1
  name hdr-rs1-r1-t1
  then service-data-flow-id payload 1300
  then protocol-inspection sip-rule-set sip-rs1
  from network-port [ 5060 ]
 !
 term 2
  name hdr-rs1-r1-t2
  then protocol-inspection rtsp-rule-set rtsp-rs1
  from network-address [ 10.170.0.5 ]
 !
!
```

Example 3 Configuring a Fallback SDF-ID in a Header Rule Term

3.2 Configure Header Rule Set

A header rule set can contain one or several header rules. The header rules are evaluated in ascending order by a user-defined `rule-id`. The `rule-id` is a numerical value from 1 through 999,999.

To add a header rule to a header rule set, include the following statement:

```
Ericsson(config)# epg pgw service-identification header
-rule-set <name> rule <rule-id>
   name <rule-name>
```

If a header rule set includes multiple header rules that classify unsolicited traffic, these header rules must be configured sequentially with highest priorities. Otherwise an error message is returned. Each of these header rules takes the priority according to its position in the configuration file. For more information, see the recommendation "Group and configure first the header rules that are not associated with application layer protocol inspection" in SACC Optimization.

3.3 Associate Header Rule Set with Service Set

One or more header rule sets can be associated with a service set.

To associate one or more header rule sets with a service set, include the following statement:

```
Ericsson(config)# epg pgw service-set <service-set-name> serv
ice-identification header-rule-sets <name>
   priority <rule-set-priority>
```

Note: The `priority` attribute is used to sort the configured rule sets list, so that the rule sets are chosen according to the ordered list.

The header rule set containing header rules that classify unsolicited traffic must be configured with the highest priority in the service set. Otherwise an error message is returned. For more information, see the corresponding recommendations in SACC Optimization.

3.4 Configure IPv6 Address Normalization

To enable IPv6 address normalization on node level in the PISC configuration, include the following statement:

```
Ericsson(config)# epg pgw service-identification
   enable-ipv6-address-normalization
```

To enable IPv6 address normalization for a service set, include the following statement:

```
Ericsson(config)# epg pgw service-set <service-set-na
me> service-identification
   enable-ipv6-address-normalization
```

IPv6 address normalization is disabled by default.

When configuring the parameters listed below, it is mandatory to enclose an IPv6 address in square brackets and to enclose the complete string in double quotation marks (for example, `"[2000:3ffe:1::]"`):

— `text-rule` for URI host normalization

— `uri`, `host`, and `extended-header` in an HTTP-WSP rule

— `any-mms-destination` in an `mms-send`

— `mm-origin` in an `mms-retrieve`

— `sender` in an SMTP rule

— `answer-name` in a DNS rule

For more information on IPv6 address normalization, refer to Packet Inspection and Service Classification (PISC).

3.5 Configure SDF-IDs for Out-of-Order Fragmented IP Packets

It is possible to configure in the service set an SDF-ID for the out-of-order fragmented IP packets arriving before the first packet and not matching any rule.

To configure the SDF-ID for the out-of-order fragmented IP packets, include the following statement:

```
Ericsson(config)# epg pgw service-set <service-set-na
me> service-identification
   out-of-order-fragmented-ip-sdf-id <sdf-id>
```

For more information on out-of-order fragmented IP packets, refer to Packet Inspection and Service Classification (PISC).

3.6 Enable RTCP Traffic Detection

To enable RTCP traffic detection, include the following statement:

```
Ericsson(config)# epg pgw service-set <service-set-na
me> service-identification
   enable-rtcp-detection
```

This configuration is valid for all the rules and rule sets in the service set. See Section 4.10.1.2 on page 45 and Section 4.11.1.2 on page 49 for how to configure the respective rules to separate RTP and RTCP contents from RTSP and SIP traffic.

RTSP or SIP rules and rule sets in a service set can be reused in other service sets only if `enable-rtcp-detection` is configured in all the relevant service sets.

If `enable-rtcp-detection` is not configured, RTCP traffic is classified in the default payload for the service set.

3.7 Configure Classification of Retransmitted TCP Packets and Duplicated TCP ACK Packets

The GGSN and PGW support classification of retransmitted TCP packets and duplicated TCP ACK packets into one SDF-ID or separate SDF-IDs defined at service set level.

Note: To configure classification of retransmitted TCP packets, exclusion of retransmitted packets from measured volume must not be configured for retransmitted TCP packets or duplicated TCP packets. For more information about excluding retransmitted packets from measured volume, refer to Charging Methods.

3.7.1 Configure Classification of Retransmitted TCP Packets

To configure an SDF-ID for classification of retransmitted TCP packets, include the following statement:

```
Ericsson(config)# epg pgw service-set <service-set-name> service-i
dentification tcp-retransmission-handling retransmission-reporting
   payload <sdf-id>
```

3.7.2 Configure Classification of Duplicated TCP ACKs

To configure an SDF-ID for classification of duplicated TCP ACK packets, include the following statement:

```
Ericsson(config)# epg pgw service-set <service-set-name> service-i
dentification tcp-retransmission-handling duplicate-ack-reporting
   payload <sdf-id>
```

4 Configure Deep Packet Inspection

To configure DPI of an application layer protocol, the following actions are mandatory:

— Configure a DPI rule. A rule consists of one or several terms. The terms are evaluated in descending order by a user-defined `term-id`. The `term-id` is a numerical value from 1 to 999,999. The EPG capacity is only affected by the total number of terms, refer to EPG Characteristics.

  To configure a term in a DPI rule, take the following actions:

  - Configure the match conditions. If several match conditions are configured in a term, all conditions must be fulfilled for the term to match. Multiple match conditions in the same term are handled in a logical `AND` relation.
  - Configure the unique resulting SDF-ID.

— Configure a DPI rule set.

— Associate the DPI rule set with a header rule. A header rule can only have one associated DPI rule set.

Note: Do not associate the DPI rule set with a header rule that classifies unsolicited traffic. This is because the main purpose of unsolicited traffic classification is to block the matched traffic regardless of the application layer protocol.

The following sections describe how to perform these steps for each supported protocol. It also describes other general configuration of DPI.

4.1 Special Characters in Terms

When configuring DPI terms, use quotation marks around the whole string if the term string includes any character other than A-Z, a-z, or 0–9, see Example 4.

The following characters are supported: A-Z, a-z, 0–9, -, ., _, ~, :, /, ?, #, [,], @, !, $, &, ', (,), *, +, ,, ;, =, %, |, ^, {, }, \, <space>.

Note: A backslash, "\", must be entered with two backslashes: "\\".

Warning!

When entering the DPI term using double quotes, the line is interpreted as a string. It may be possible to include other characters than those described above, such as language-specific characters. In such cases, the operation of the DPI term may be other than expected.

```
epg pgw service-identification http-wsp-rule pr2
 term 1
  name t1
  from uri is http://myweb.com/useQuotation?opt=5&set=mine
 !
!
```

Example 4 Using Quotation Marks for DPI Terms

4.2 Configure Textual Matching

This section describes how to configure text rules and text rule sets for textual matching.

4.2.1 Configure Text Rule

A text rule consists of one or more terms. The terms are evaluated in descending order by a user-defined `term-id`. The `term-id` is a numerical value from 1 to 999,999. Each term consists of one or several match conditions.

To configure textual matching for a term, include the following statement:

```
Ericsson(config)# epg pgw service-identification text-r
ule <rule-name> term <term-id>
   name <term-name>
   is <string>
   not-is <string>
   starts-with <string>
   not-starts-with <string>
   ends-with <string>
   not-ends-with <string>
   contains <string>
   not-contains <string>
```

Note: The maximum number of match conditions that can be configured is 10.

Strings are not case-sensitive by default. For information on maximum number of string characters supported, refer to EPG Characteristics.

Wildcard characters are partially supported for textual matching configuration. For more information on how to use wildcard characters in text rules, refer to Packet Inspection and Service Classification (PISC).

In Example 5, both conditions `ends-with` and `starts-with` have to be fulfilled for the term `t1` to match.

```
epg pgw service-identification http-wsp-rule dpi1
 term 1
  name t1
  from uri starts-with platinum
  from uri ends-with premium
 !
!
```

Example 5 Multiple Match Conditions

Multiple strings in the square brackets are handled as separate parameters of separate match conditions in a logical `AND` relation.

In Example 6, all the comma-separated string parameters `bronze`, `silver`, and `gold` have to be present to fulfill the match condition `contains`.

```
epg pgw service-identification http-wsp-rule dpi1
 term 1
  name t1
  from uri contains [ bronze silver gold ]
 !
!
```

Example 6 Matching Multiple Strings

In Example 7, the match condition `notEndsWith` is fulfilled if a URI does not end with `premium` and `platinum`.

```
epg pgw service-identification http-wsp-rule dpi1
 term 1
  name t1
  from uri not-ends-with [ platinum premium ]
 !
!
```

Example 7 Not Matching Multiple Strings

To perform logical `OR` operations, the comma-separated strings in the square brackets must be split into separate terms.

4.2.2 Configure Text Rule Set

A text rule set consists of one or more text rules. The text rules are evaluated in descending order by a user-defined `rule-id`. To add a text rule to a text rule set, include the following statement:

```
Ericsson(config)# epg pgw service-identification text-
rule-set <name> rule <rule-id>
   name <text-rule-name>
```

4.3 Configure Classification of Empty UDP Packets

To configure an SDF-ID for classification of empty UDP packets, include the following statement:

```
Ericsson(config)# epg pgw service-set <service-set-name
> service-identification no-udp-payload
   payload <sdf-id>
```

4.4 Configure DNS Inspection

This section describes configuration of DNS inspection, see also Section 4 on page 14.

Note: Only DPI of DNS over UDP is supported.

4.4.1 Configure DNS Rule

This section describes how to configure a rule for packet matching based on the DNS protocol.

4.4.1.1 Configure Match Conditions

This section describes how to configure match conditions for a term in a DNS rule.

Query Name

To configure packet matching based on query name, include the following statement:

```
Ericsson(config)# epg pgw service-identification dns-rule
<rule-name> term <term-id> from dns query-name
   is <string>
   not-is <string>
   starts-with <string>
   not-starts-with <string>
   ends-with <string>
   not-ends-with <string>
   contains <string>
   not-contains <string>
```

Note: The maximum number of match conditions that can be configured is 10.

Answer Name

To configure packet matching based on answer name, include the following statement:

```
Ericsson(config)# epg pgw service-identification dns-rule
<rule-name> term <term-id> from dns answer-name
   contains <string>
   not-contains <string>
   case
```

By default, case is not considered. The optional `case` statement toggles sensitivity to case.

4.4.1.2 Configure Resulting SDF-ID

To configure the SDF-ID to assign to traffic that matches the conditions in a term, include the following statement:

```
Ericsson(config)# epg pgw service-identification dns-rule
<rule-name> term <term-id> then
   payload <sdf-id>
```

4.4.2 Configure DNS Rule Set

A rule set can contain one or several rules. The rules are evaluated in descending order by a user-defined `rule-id`. The `rule-id` is a numerical value from 1 through 999,999.

To add a rule to a rule set, include the following statement:

```
Ericsson(config)# epg pgw service-identification dns-r
ule-set <name> rule <rule-id>
   name <text-rule-name>
```

4.4.3 Associate DNS Rule Set with Header Rule

To associate a rule set with a header rule, include the following statement:

```
Ericsson(config)# epg pgw service-identification header-rule
<rule-name> term <term-id> then protocol-inspection
   dns-rule-set <name>
```

4.4.4 Configure DNS Cache

To configure the DNS cache, the following prerequisites apply:

— The DNS spoofing protection must be enabled. See Section 4.4.5 on page 20.

— A list of trusted DNS servers must be configured. This list is stored by default in the DNS cache. See Section 3.1.2.1 on page 9.

A minimum Time to Live (TTL) value can be configured, if the TTL in the response is less than the minimum TTL value. See Section 3.1.2.2 on page 9.

Note: The GGSN and PGW support runtime configuration of the DNS cache. Configuration changes take effect immediately and previously cached records are deleted.

4.4.4.1 Enable and Disable the DNS Cache

The DNS cache must be enabled at service-set level. To enable the DNS cache at service-set level, include the following statement:

```
Ericsson(config)# epg pgw service-set <name> serv
ice-identification
   enable-dns-cache
```

4.4.4.2 Configure a DNS Cache Whitelist

A DNS cache whitelist is a list of DNS domain names that can be cached by the PGW. When configured, the PGW caches only the DNS records for which the domain name contains one of the strings configured in the whitelist. The domain

names are compared exactly with the ones defined in the whitelist, therefore wildcards are not allowed. To configure a DNS cache whitelist, include the following statement once per domain name to be added to the list:

```
Ericsson(config)# epg pgw service-identification dns-monitoring dns-cache
   whitelist-name <name>
```

```
Ericsson(config)# epg pgw service-identification dns-monitoring dns-cache white
Ericsson(config)# epg pgw service-identification dns-monitoring dns-cache white
Ericsson(config)# epg pgw service-identification dns-monitoring dns-cache white
```

Example 8 Configuring a Whitelist for DNS Cache

4.4.4.3 Configure a DNS Cache Blacklist

A DNS cache blacklist is a list of DNS domain names that must not be cached by the PGW. When configured, the PGW does not cache the DNS records for which the domain name contains one of the strings configured in the blacklist. The domain names are compared exactly with the ones defined in the blacklist, therefore wildcards are not allowed. To configure a DNS cache blacklist, include the following statement once per domain name to be added to the list:

```
Ericsson(config)# epg pgw service-identification dns-monitoring dns-cache
   blacklist-name <name>
```

```
Ericsson(config)# epg pgw service-identification dns-monitoring dns-cache black
Ericsson(config)# epg pgw service-identification dns-monitoring dns-cache black
Ericsson(config)# epg pgw service-identification dns-monitoring dns-cache black
```

Example 9 Configuring a Blacklist for DNS Cache

4.4.4.4 Configure Maximum Number of Records in the DNS Cache

To configure a maximum number of records to store per DNS cache, include the following statement:

```
Ericsson(config)# epg pgw service-identification dns-monitoring dns-cache
   max-entries-name <number>
```

The value range of `max-entries-name` is 1–4,294,967,296. The default value is 100,000, which applies to all DNS caches.

4.4.5 Configure DNS Spoofing Protection

By default, the DNS spoofing protection is disabled.

To enable the DNS spoofing protection, include the following statement:

```
Ericsson(config)# epg pgw service-identification dns-monitoring
   enable-spoofing-protection
```

4.5 Configure FTP Inspection

This section describes configuration of File Transfer Protocol (FTP) inspection, see also Section 4 on page 14.

4.5.1 Configure FTP Rule

This section describes how to configure a rule for packet matching based on the FTP.

4.5.1.1 Configuring Match Conditions

This section describes how to configure match conditions for a term in an FTP rule.

Filename

To configure packet matching based on filename, include the following statement:

```
Ericsson(config)# epg pgw service-identification ftp-rule
<rule-name> term <term-id> from ftp filename
   contains <string>
   not-contains <string>
   case
```

By default, case is not considered. The optional `case` statement toggles sensitivity to case.

FTP Operations

To configure packet matching based on FTP operations, include one or several of the following statements:

```
Ericsson(config)# epg pgw service-identification ftp-rule
<rule-name> term <term-id> from ftp
   operation (retrieve | store)
```

4.5.1.2 Configure Resulting SDF-ID

To configure the SDF-ID to assign to traffic that matches the conditions in a term, include the following statement:

```
Ericsson(config)# epg pgw service-identification ftp-rule
<rule-name> term <term-id> then
   payload <sdf-id>
```

4.5.2 Configure FTP Rule Set

A rule set can contain one or several rules. The rules are evaluated in descending order by a user-defined `rule-id`. The `rule-id` is a numerical value from 1 through 999,999.

To add a rule to a rule set, include the following statement:

```
Ericsson(config)# epg pgw service-identification ftp-r
ule-set <name> rule <rule-id>
   name <ftp-rule-name>
```

4.5.3 Associate FTP Rule Set with Header Rule

To associate a rule set with a header rule, include the following statement:

```
Ericsson(config)# epg pgw service-identification header-rule
<rule-name> term <term-id> then protocol-inspection
   ftp-rule-set <name>
```

4.5.4 Enable FTP Path Tracking

To enable FTP path tracking for a service set, include the following statement:

Note: FTP path tracking is resource consuming and can have an impact on the capacity and performance of the EPG.

```
Ericsson(config)# epg pgw service-set <service-set-na
me> service-identification
   enable-ftp-path-tracking
```

This configuration applies to an entire service set.

4.6 Configure HTTP, WSP, and MMS Inspection

This section describes configuration of HTTP, WSP, and MMS inspection, see also Section 4 on page 14.

Inspection of these protocols can be configured in a shared rule with separate match conditions leading to the same SDF-ID.

4.6.1 Configure HTTP-WSP Rule

This section describes how to configure a rule for packet matching based on the HTTP, WSP, and Multimedia Messaging Service (MMS) protocols.

4.6.1.1 Configuring URI Match Conditions

To configure packet matching based on URI, include the following statement:

```
Ericsson(config)# epg pgw service-identification http-wsp-rule
<rule-name> term <term-id> from uri
   include-uri-handling
      case
   is <string>
   not-is <string>
   starts-with <string>
   not-starts-with <string>
   ends-with <string>
   not-ends-with <string>
   contains <string>
   not-contains <string>
```

Note: The maximum number of match conditions that can be configured is 10.

By default, case is not considered. The optional `case` statement toggles sensitivity to case.

If a URI filter only applies to one protocol, it is recommended to add a protocol filter for improved performance.Example 10 shows a URI filter only applicable for HTTP traffic.

```
epg pgw service-identification http-wsp-rule httpRule1
 term 1
  name term1
  then payload 5
  from uri starts-with http://ericsson.com
  from http
 !
!
```

Example 10 Configuring a URI Filter for HTTP Traffic

Enabling Escape Character Conversion

To enable conversion of escape characters before evaluation of HTTP URIs, enter the following:

```
Ericsson(config)# epg pgw service-set <service-set-name> serv
ice-identification unescape-conversion
   http
```

To enable conversion of escape characters before evaluation of WSP URIs, enter the following:

```
Ericsson(config)# epg pgw service-set <service-set-name> serv
ice-identification unescape-conversion
```

```
   wsp
```

This configuration is valid for all DPI rules inspecting HTTP or WSP traffic in the service set. Conversion of escape characters can also be enabled for all DPI rules in the EPG, see Section 4.15 on page 60.

4.6.1.2 Configure HTTP Match Conditions

This section describes how to configure match conditions for HTTP in an HTTP-WSP rule term.

HTTP Host

To configure packet matching based on HTTP host, include the following statement:

```
Ericsson(config)# epg pgw service-identification http-wsp-rule
<rule-name> term <term-id> from http header host
   case
   is <string>
   not-is <string>
   starts-with <string>
   not-starts-with <string>
   ends-with <string>
   not-ends-with <string>
   contains <string>
   not-contains <string>
```

Note: The maximum number of match conditions that can be configured is 10.

By default, case is not considered. The optional `case` statement toggles sensitivity to case.

There is a limitation of HTTP analysis when analyzing the following URL formats:

— `http://USERNAME@HOST/DIR/FILE.EXT`

— `http://USERNAME:PASSWORD@HOST/DIR/FILE.EXT`

The DPI engine by default interprets `USERNAME` as the hostname, when in fact the hostname is `HOST`. For example, `http://operator.com@othersite.com` is classified under `http://operator.com` and not as `http://othersite.com`, leading to a potential wrong classification.

HTTP Operations

To configure packet matching based on HTTP operations, include one or several of the following statements:

```
Ericsson(config)# epg pgw service-identification http-wsp-rule
<rule-name> term <term-id> from http
```

```
   operation (connect | delete | get | head | options
| post | put | trace)
```

Content Type

To configure packet matching based on content type, include the following statement:

```
Ericsson(config)# epg pgw service-identification http-wsp-rule
<rule-name> term <term-id> from http content-type
   case
   is <string>
   not-is <string>
   starts-with <string>
   not-starts-with <string>
   ends-with <string>
   not-ends-with <string>
   contains <string>
   not-contains <string>
```

Note: The maximum number of match conditions that can be configured is 10.

By default, case is not considered. The optional `case` statement toggles sensitivity to case.

Extended Headers

For information on configuring packet matching based on extended headers, see Section 4.6.8 on page 32.

HTTP Domain

To configure packet matching based on HTTP domain, include the following statement:

```
Ericsson(config)# epg pgw service-identification http-wsp-rule
<rule-name> term <term-id> from domain
   is <string>
   not-is <string>
   starts-with <string>
   not-starts-with <string>
   ends-with <string>
   not-ends-with <string>
   contains <string>
   not-contains <string>
   domain-length <number>
```

Note: The maximum number of match conditions that can be configured is 10.

It is possible to include HTTP domain together with other match conditions in a term. Only one HTTP domain statement is allowed in a term.

Wildcard characters are partially supported for textual matching configuration. For more information on how to use wildcard characters in HTTP domain configuration, refer to Packet Inspection and Service Classification (PISC).

HTTP Response Codes

To configure packet matching based on HTTP response codes, include the following statement:

```
Ericsson(config)# epg pgw service-identification http-wsp-rule
<rule-name> term <term-id> from http response-code
   is (<value> | <value>-<value>)
```

The configurable value range is 100–999. HTTP response codes from 100 to 399 are considered successful and an HTTP success event is triggered if event tracking is activated. HTTP response codes from 400 to 999 are considered as error codes and an error event is triggered if event tracking is activated. For more information on HTTP events, refer to Charging Methods.

Note: Response codes are matched only in HTTP responses, not in requests.

Response codes can be configured with or without event tracking configured in the rule. If event tracking is configured, it is only allowed to set event tracking to `response`. Event tracking configured as `start` or `complete` does not work with response codes as the match condition for HTTP in an HTTP-WSP rule. For more information on how to activate event tracking, refer to Charging Methods Configuration.

HTTP Midflows

To configure packet matching based on HTTP midflows, include the following statement:

```
Ericsson(config)# epg pgw service-identification http-wsp-rule
<rule-name> term <term-id> from http
   mid-flow
```

Segmented HTTP Messages

To enable the classification of segmented HTTP messages, include the following statement:

```
Ericsson(config)# epg pgw service-identification http-wsp-rule
<rule-name> term <term-id> from http
   is-segmented
```

Unsegmented HTTP Messages

To enable the classification of unsegmented HTTP messages, include the following statement:

```
Ericsson(config)# epg pgw service-identification http-wsp-rule
<rule-name> term <term-id> from http
   not-is-segmented
```

HTTP Out-of-Order Segments

To configure packet matching for TCP out-of-order uplink packets, include the following statement:

```
Ericsson(config)# epg pgw service-identification http-wsp-rule
<rule-name> term <term-id> from http
   uplink-out-of-order
```

4.6.1.3 Configure WSP Match Conditions

This section describes how to configure match conditions for WSP in an HTTP-WSP rule term.

WSP Operations

To configure packet matching based on WSP operations, include one or several of the following statements:

```
Ericsson(config)# epg pgw service-identification http-wsp-rule
<rule-name> term <term-id> from wsp
   operation (connect | delete | get | head | options
| post | put | trace)
```

Content Type

To configure packet matching based on content type, include the following statement:

```
Ericsson(config)# epg pgw service-identification http-wsp-rule
<rule-name> term <term-id> from wsp content-type
   case
   is <string>
   not-is <string>
   starts-with <string>
   not-starts-with <string>
   ends-with <string>
   not-ends-with <string>
   contains <string>
   not-contains <string>
```

Note: The maximum number of match conditions that can be configured is 10.

By default, case is not considered. The optional `case` statement toggles sensitivity to case.

Extended Headers

For information on configuring packet matching based on extended headers, see Section 4.6.8 on page 32.

4.6.1.4 Configure MMS Match Conditions

This section describes how to configure match conditions for MMS traffic in an HTTP-WSP rule term. The payload packets belonging to the MMS body can be captured based on the MMS operation. The payload packets containing only the HTTP/WSP header of an MMS message can be captured with a general rule for MMS traffic.

Note: Inspection of MMS traffic is resource consuming and can have an impact on the capacity and performance of the EPG.

Send Operation

To configure packet matching based on the MMS send operation, include the following statement:

```
Ericsson(config)# epg pgw service-identification http-wsp-rule
<rule-name> term <term-id> from mms mms-send any-mms-destination
   case
   contains <string>
   not-contains <string>
```

Using the `any-mms-destination` statement to match only specified destinations is optional. By default, case is not considered. The optional `case` statement toggles sensitivity to case.

Retrieve Operation

To configure packet matching based on the MMS retrieve operation, include the following statement:

```
Ericsson(config)# epg pgw service-identification http-wsp-rule
<rule-name> term <term-id> from mms mms-retrieve mm-origin
   case
   is <string>
   not-is <string>
   starts-with <string>
   not-starts-with <string>
   ends-with <string>
   not-ends-with <string>
   contains <string>
   not-contains <string>
```

Note: The maximum number of match conditions that can be configured is 10.

Using the `mm-origin` statement to match only specified originators is optional. By default, case is not considered. The optional `case` statement toggles sensitivity to case.

Notification Operation

To configure packet matching based on the MMS notification operation, include the following statement:

```
Ericsson(config)# epg pgw service-identification http-wsp-rule
<rule-name> term <term-id> from mms
   mms-notification
```

Forward Operation

To configure packet matching based on the MMS forward operation, include the following statement:

```
Ericsson(config)# epg pgw service-identification http-wsp-rule
<rule-name> term <term-id> from mms
   mms-forward
```

Acknowledge Operation

To configure packet matching based on the MMS acknowledge operation, include the following statement:

```
Ericsson(config)# epg pgw service-identification http-wsp-rule
<rule-name> term <term-id> from mms
   mms-acknowledge
```

Read-Report Operation

To configure packet matching based on the MMS read-report operation, include the following statement:

```
Ericsson(config)# epg pgw service-identification http-wsp-rule
<rule-name> term <term-id> from mms
   mms-read-report
```

General

To capture all MMS traffic, including the payload packets containing only the HTTP/WSP header of MMS messages, include the following statement:

```
Ericsson(config)# epg pgw service-identification http-wsp-rule
<rule-name> term <term-id> from
   mms
```

4.6.1.5 Configure Resulting SDF-ID

To configure the SDF-ID to assign to traffic that matches the conditions in a term, include the following statement:

```
Ericsson(config)# epg pgw service-identification http-wsp-rule
<rule-name> term <term-id> then
   payload <sdf-id>
```

4.6.2 Configure HTTP-WSP Rule Set

A rule set can contain one or several rules. The rules are evaluated in descending order by a user-defined `rule-id`. The `rule-id` is a numerical value from 1 through 999,999.

To add a rule to a rule set, include the following statement:

```
Ericsson(config)# epg pgw service-identification http-ws
p-rule-set <name> rule <rule-id>
   name <rule-name>
```

4.6.3 Associate HTTP-WSP Rule Set with Header Rule

To associate a rule set with a header rule, include the following statement:

```
Ericsson(config)# epg pgw service-identification header-rule
<rule-name> term <term-id> then protocol-inspection
   http-wsp-rule-set <name>
```

Filtering HTTP, WSP, and MMS requires considerable processing resources. It is mandatory to differentiate the traffic in the associated header rules, for example by dividing TCP and UDP traffic into two or more header rules, to decrease the impact on EPG capacity.

If DPI for MMS is configured, it is recommended to use a separate header rule for only MMS DPI rules with the IP address filter set to MMS Center (MMSC) addresses instead of using a header rule for MMS DPI rules that is also used for HTTP and WSP DPI rules. For more information, refer to SACC Optimization.

4.6.4 Enable HTTP Pipelining Support

To enable the support for inspection of pipelined HTTP requests for a service set, include the following statement:

Note: Support for pipelined HTTP requests is resource consuming and can have an impact on the capacity and performance of the EPG.

```
Ericsson(config)# epg pgw service-set <service-set-na
me> service-identification
```

```
   enable-http-pipeline
```

4.6.5 Enable Reassembly of WTP Fragments

To enable reassembly of Wireless Transaction Protocol (WTP) fragments for a service set, include the following statement:

Note: Reassembly of WTP fragments is resource consuming and can have an impact on the capacity and performance of the EPG.

```
Ericsson(config)# epg pgw service-set <service-set-name> serv
ice-identification enable-wtp-reassembly
   max-reassembled-bytes <bytes>
   max-reassembled-packets <packets>
```

Configuring the maximum number of bytes to reassemble is optional. The configurable range is 0–362,100. The default value is 0, which means there is no limit.

Configuring the maximum number of packets to reassemble is optional. The configurable range is 0–255. The default value is 3. Setting the value to 0 means that there is no limit.

4.6.6 Configure HTTP Header Enrichment and Override

For information how to configure content enrichment, refer to Content Enrichment Configuration.

4.6.7 Configure HTTP Request Header Deferred Charging

To enable HTTP request header deferred charging, include the following statement:

```
Ericsson(config)# epg pgw service-set <service-set-name> service-
identification
   http-deferred-charging-sdf-id <sdf-id>
```

The deferred charging SDF-ID is used to classify the fragments after receiving the end of the HTTP request. If the request is completed, the SDF-ID volume is reported within the first HTTP/WSP matching an authorized SDF-ID with HTTP deferred charging reporting enabled. If the request fails, that is, the request is not completed because of connection setup failure, the volume is not reported.

The SDF-ID must be a number between 0 and 2^{32}-1.

HTTP request header deferred charging is disabled by default.

When HTTP request header deferred charging is enabled in a service set, the following applies:

— HTTP-WSP rules and rule sets in the service set may only be reused in different service sets as long as a common feature set applies.

— Use the same header rule SDF-ID for all header rules pointing to the same HTTP-WSP rule set.

For more information on HTTP request header deferred charging, refer to Packet Inspection and Service Classification (PISC).

4.6.8 Configure Packet Matching Based on Extended HTTP Headers

To configure packet matching based on extended HTTP headers, do the following:

1. Configure an extended header.
2. Configure an HTTP-WSP rule using the extended header.

The following sections describe steps 1 and 2. A complete example is given at the end of the section.

Note: A maximum of three extended headers can be configured per service set; two with textual content and one with decimal content.

4.6.8.1 Configure Extended Headers

It is possible to configure extended headers on the service identification level. The header-name must be composed of printable ASCII characters (from 33 through 126).

Note: Configuring extended HTTP headers increases resource consumption and can have an impact on the capacity and performance of the EPG.

To configure packet matching based on an extended HTTP header with textual content, include the following statement:

```
Ericsson(config)# epg pgw service-identification extended-header-definition htt
   text-content <header-name>
```

To configure packet matching based on an extended HTTP header with decimal content, include the following statement:

```
Ericsson(config)# epg pgw service-identification extended-header-definition htt
   decimal-content <header-name>
```

To configure packet matching based on an extended WSP header with textual content, include the following statement:

```
Ericsson(config)# epg pgw service-identification extended-header-definition wsp
   text-content <header-name>
```

4.6.8.2 Configure Rule Using Extended Headers

Packet matching based on extended headers can be added as part of an HTTP-WSP rule and term already configured according to the instructions in Section 4.6.1.2 on page 24 or Section 4.6.1.3 on page 27. It can also be configured as a separate HTTP-WSP rule or term.

To configure a rule matching the textual content of an extended HTTP header, include the following statement:

```
Ericsson(config)# epg pgw service-identification http-w
sp-rule <rule-name> term <term-id> from http extended-h
eader text-content <header-name>
   case
   is <string>
   not-is <string>
   starts-with <string>
   not-starts-with <string>
   ends-with <string>
   not-ends-with <string>
   contains <string>
   not-contains <string>
```

Note: The maximum number of match conditions that can be configured is 10.

To configure a rule matching the textual content of an extended WSP header, include the following statement:

```
Ericsson(config)# epg pgw service-identification http-w
sp-rule <rule-name> term <term-id> from wsp extended-he
ader text-content <header-name>
   case
   is <string>
   not-is <string>
   starts-with <string>
   not-starts-with <string>
   ends-with <string>
   not-ends-with <string>
   contains <string>
   not-contains <string>
```

Note:

- The maximum number of match conditions that can be configured is 10.
- An extended HTTP header with decimal content cannot be used for classification.

4.6.9 Configure Extended HTTP Header Authentication for Sponsored Data

The following extended HTTP header names can be configured for sponsored data, in addition to the extended HTTP header names described in Section 4.6.8.1 on page 32:

— Service name (`x-content-id`), that can be configured in an HTTP-WSP rule and is used for packet matching.

— Authentication information (`x-auth-info`), that is used for MD5 algorithm authentication.

— Time stamp (`x-date`), that is used for time stamp authentication.

For more information on extended HTTP header authentication, refer to Packet Inspection and Service Classification (PISC)

To configure packet matching based on an extended HTTP header with textual content, enter the following:

```
Ericsson(config)# epg pgw service-identification extended-header-definition htt
   text-content <header-name>
```

To configure packet matching based on an extended HTTP header with date content, enter the following:

```
Ericsson(config)# epg pgw service-identification extended-header-definition htt
   date-content <header-name>
```

Example 11 shows an example how to configure header names for extended HTTP header authentication.

4.6.9.1 Configure Authentication Profiles

To configure authentication profiles, include the following statement:

```
Ericsson(config)# epg pgw service-identification auth
-content-profile <profile-id>
   enable-algorithm
   enable-threshold
   time-valid-diff <number>
```

MD5 algorithm authentication and time stamp authentication can be enabled or disabled independently per authentication profile. To disable MD5 algorithm authentication, the `enable-algorithm` attribute must not be configured in the authentication profile. To disable time stamp authentication, the `enable-threshold` attribute must not be configured in the authentication profile. If the `enable-threshold` attribute is not configured, the `time-valid-diff` attribute cannot be configured.

Note: Both types of authentication can be disabled at rule level and service set level. The authentication rules at rule level have higher priority than authentication rules at service-set level.

The default value of the `time-valid-diff` attribute is 1 second. The supported range of configurable values for this attribute is 1–600 seconds.

Note: If HTTP response classification is enabled, it is recommended to use a higher value than 1 for the `time-valid-diff` attribute.

Example 11 shows an example how to configure an authentication profile.

4.6.9.2 Associate Default Authentication Profile with Service Set

To associate the default authentication profile with a service set, include the following statement:

```
Ericsson(config)# epg pgw service-set <service-set-na
me> service-identification
   default-auth-content-profile <profile-id>
```

The value of the `default-auth-content-profile` attribute must be the same as the one that identifies an `auth-content-profile` class.

If an HTTP-WSP rule with an `auth-rule` is configured, an authentication profile must be configured and it must be associated with a service set as default authentication profile.

Example 11 shows an example how to associate the default authentication profile with a service set.

4.6.9.3 Configure Authentication Rules

To configure authentication rules, include the following statement:

```
Ericsson(config)# epg pgw service-identification auth-con
tent-rule <rule-id> extended-header
   name <string>
   is
      auth-field <field-id>
         constant-field <string>
         extended-header-name <string>
         name <string>
```

Example 11 shows an example how to configure an authentication rule.

4.6.9.4 Associate Authentication Profile with Authentication Rule

To associate an authentication profile with an authentication rule, include the following statement:

```
Ericsson(config)# epg pgw service-identification
auth-content-rule <rule-id>
   auth-content-profile-name <profile-id>
```

The value of the `auth-content-profile-name` attribute must be the same as the one that identifies an `auth-content-profile` class.

If an authentication rule is configured, it must be associated with an authentication profile. If none has been configured in the authentication rule, the default authentication profile applies.

Example 11 shows an example how to associate an authentication profile with an authentication rule.

4.6.9.5 Associate Authentication Rule with HTTP-WSP Rule

To associate an authentication rule with an HTTP-WSP rule, include the following statement:

```
Ericsson(config)# epg pgw service-identification http-wsp-rule
<rule-name> term <term-id> from http
   auth-rule-name <rule-name>
```

The value of the `auth-rule-name` attribute must be the same as the one that identifies an `auth-content-rule` class.

Example 11 shows an example how to associate an authentication rule with an HTTP-WSP rule.

4.6.9.6 Configuration Example for Extended HTTP Header Authentication

Example 11 shows an example of the extended HTTP header authentication.

```
Associate the profA default authentication profile
with the ss1 service set:

epg pgw service-set ss1
 service-identification service-data-flow-id default payload 1
 service-identification default-auth-content-profile profA
 service-identification header-rule-sets hrs
  priority 999999
 !
!
Configure an extended HTTP header with x-content-id, x-auth-info, and x-date

epg pgw service-identification extended-header-definition http text-content
epg pgw service-identification extended-header-definition http date-content

Configure the hrs header rule set:

epg pgw service-identification header-rule-set hrs
 rule 1
  name hr
 !
!

Configure the hr header rule:

epg pgw service-identification header-rule hr
 term 1
  name term1
  then protocol-inspection http-wsp-rule-set hwrs
  from network-port [ 80 ]
  from protocol tcp
 !
!

Configure the hwrs http-wsp rule set:

epg pgw service-identification http-wsp-rule-set hwrs
 rule 1
  name hwr
 !
!

Configure the hwr http-wsp rule and associate the authruleA authentication r

epg pgw service-identification http-wsp-rule hwr
 term 1
  name auth-term
  then payload 900
  from http auth-rule-name authruleA
  from http extended-header text-content name x-content-id
  from http extended-header text-content is value
  !
 !
!

Configure the profA authentication profile:

epg pgw service-identification auth-content-profile profA
 enable-algorithm
 enable-threshold
```

Example 11 Extended HTTP Header Authentication Example

4.6.10 Configure URI Host Normalization

To normalize non-standard URLs in HTTP Post requests, the approved server prefixes to execute URI host normalization need to be defined and then URI host normalization must be enabled.

To define the server prefixes on node level using the `text-rule` command that is limited to `is` operator when used with URI host normalization, include the following statement:

```
Ericsson(config)# epg pgw service-identification text-r
ule <rule-name> term <term-id>
   name <term-name>
   is <string>
```

To add the text rule to the text rule set, include the following statement:

```
Ericsson(config)# epg pgw service-identification text-rule-set
<rule-set-name> rule <rule-id>
   name <text-rule-name>
```

To enable URI host normalization by including a reference to the text rule set that includes the server prefixes, include the following statement:

```
Ericsson(config)# epg pgw service-set <service-set-name> serv
ice-identification uri-host-normalization
   hosts <text-rule-set-name>
```

```
epg pgw service-identification text-rule tr
 term 1
  name t1
  is   1st.firstserver.com
 !
 term 2
  name t2
  is   2nd.firstserver.com
 !
!
epg pgw service-identification text-rule-set trs
 rule 1
  name tr
 !
!
epg pgw service-set ss
 service-identification uri-host-normalization hosts trs
!
```

Example 12 Normalizing Non-Standard URLs in HTTP Post Requests

4.7 Configure IMAP Inspection

This section describes the configuration of Internet Message Access Protocol (IMAP) inspection, see also Section 4 on page 14.

4.7.1 Configure IMAP Rule

This section describes how to configure a rule for packet matching based on IMAP.

4.7.1.1 Configure Match Conditions

This section describes how to configure match conditions for a term in an IMAP rule.

All IMAP Packets

Configure packet matching based on IMAP by including the following statement:

```
Ericsson(config)# epg pgw service-identification imap-rule
<rule-name> term <term-id> from
   imap
```

This matches all incoming traffic including both requests and responses.

IMAP Operations

Configure packet matching based on the IMAP `LOGIN` command by including the following statement:

```
Ericsson(config)# epg pgw service-identification imap-rule
<rule-name> term <term-id> from imap
   operation login
```

This matches the IMAP `LOGIN` command including both requests and responses.

4.7.1.2 Configure Resulting SDF-ID

Configure the SDF-ID to assign to traffic that matches the conditions in a term by including the following statement:

```
Ericsson(config)# epg pgw service-identification imap-rule
<rule-name> term <term-id> then
   payload <sdf-id>
```

4.7.2 Configure IMAP Rule Set

A rule set can contain one or several rules. These rules are evaluated in ascending order by a user-defined `rule-id`. The `rule-id` is a numerical value from 1 through 999,999.

Add a rule to a rule set by including the following statement:

```
Ericsson(config)# epg pgw service-identification imap-rule-set
<rule-set-name> rule <rule-id>
   name <rule-name>
```

4.7.3 Associate IMAP Rule Set with Header Rule

Associate a rule set with a header rule by including the following statement:

```
Ericsson(config)# epg pgw service-identification header-rule
<rule-name> term <term-id> then protocol-inspection
   imap-rule-set <rule-set-name>
```

4.7.4 Configure IMAP Content Enrichment

For information how to configure content enrichment, refer to Content Enrichment Configuration.

4.8 Configure POP3 Inspection

This section describes configuration of Post Office Protocol 3 (POP3) inspection, see also Section 4 on page 14.

4.8.1 Configure POP3 Rule

This section describes how to configure a rule for packet matching based on the POP3.

4.8.1.1 Configure Match Conditions

This section describes how to configure match conditions for a term in a POP3 rule.

User

To configure packet matching based on user, include the following statement:

```
Ericsson(config)# epg pgw service-identification pop3-rule
<rule-name> term <term-id> from pop3 user
   case
   is <string>
```

```
   not-is <string>
   starts-with <string>
   not-starts-with <string>
   ends-with <string>
   not-ends-with <string>
   contains <string>
   not-contains <string>
```

Note: The maximum number of match conditions that can be configured is 10.

By default, case is not considered. The optional `case` statement toggles sensitivity to case.

POP3 Operations

To configure packet matching based on POP3 operations, include one or several of the following statements:

```
Ericsson(config)# epg pgw service-identification pop3-rule
<rule-name> term <term-id> from pop3
   operation (retr | top | list)
```

4.8.1.2 Configure Resulting SDF-ID

To configure the SDF-ID to assign to traffic that matches the conditions in a term, include the following statement:

```
Ericsson(config)# epg pgw service-identification pop3-rule
<rule-name> term <term-id> then
   payload <sdf-id>
```

4.8.2 Configure POP3 Rule Set

A rule set can contain one or several rules. The rules are evaluated in descending order by a user-defined `rule-id`. The `rule-id` is a numerical value from 1 through 999,999.

To add a rule to a rule set, include the following statement:

```
Ericsson(config)# epg pgw service-identification pop3-
rule-set <name> rule <rule-id>
   name <rule-name>
```

4.8.3 Associate POP3 Rule Set with Header Rule

To associate a rule set with a header rule, include the following statement:

```
Ericsson(config)# epg pgw service-identification header-rule
<rule-name> term <term-id> then protocol-inspection
```

```
   pop3-rule-set <name>
```

4.9 Configure QUIC Inspection

This section describes configuration of Quick UDP Internet Connections (QUIC), see also Section 4 on page 14.

4.9.1 Configure QUIC Rule

This section describes how to configure a rule for packet matching based on QUIC rules.

4.9.1.1 Configure Match Conditions

This section describes how to configure match conditions for a term in a QUIC rule.

Note: Strings configured in the following match conditions are not case-sensitive.

SNI

To configure packet matching based on the Server Name Indication (SNI) field of the Client Hello handshake message, include the following statements:

```
Ericsson(config)# epg pgw service-identification quic-r
ule <rule-name> term <term-id>
   name <term-name>
   from client-hello server-name-indication
      contains <string>
   then
      payload <sdf-id>
```

Example 13 shows an example configuration for a QUIC rule based on the SNI.

```
epg pgw service-identification quic-rule quic_r1
 term 1
  name quic_r1_t1
  from client-hello server-name-indication contains [ poc.quic.es ]
  then payload 100
 !
!
```

Example 13 QUIC Rule Configuration

Catch-All QUIC

To catch all QUIC traffic that does not match any configured field value of the handshake messages, include the following statement:

```
Ericsson(config)# epg pgw service-identification quic-rule
<rule-name> term <term-id> from
   any-quic
```

If the DPI QUIC catch-all rule is explicitly configured, QUIC traffic that does not match specific QUIC SNI rules does not go through heuristic packet inspection.

4.9.2 Configure QUIC Rule Set

A rule set can contain one or several rules. The rules are evaluated in descending order by a user-defined `rule-id`. The `rule-id` is a numerical value from 1 through 999,999.

To add a rule to a rule set, include the following statement:

```
Ericsson(config)# epg pgw service-identification quic-rule-set
<rule-set-name> rule <rule-id>
   name <rule-name>
```

Example 14 shows an example configuration for a QUIC rule set.

```
epg pgw service-identification quic-rule-set quic_rs1
 rule 1
  name quic_r1
 !
!
```

Example 14 QUIC Rule Set Configuration

4.9.3 Associate QUIC Rule Set with Header Rule

To associate a rule set with a header rule, include the following statement:

```
Ericsson(config)# epg pgw service-identification header-rule
<rule-name> term <term-id> then protocol-inspection
   quic-rule-set <rule-set-name>
```

Example 15 shows an example configuration to associate a QUIC rule set with a header rule.

```
epg pgw service-identification header-rule hr1
 term 3
  name quic_hr1_t1
  then service-data-flow-id payload 150
  then protocol-inspection quic-rule-set quic_rs1
 !
!
```

Example 15 Associate QUIC Rule Set with Header Rule

4.10 Configure RTSP Inspection

This section describes configuration of RTSP inspection, see also Section 4 on page 14.

4.10.1 Configure RTSP Rule

This section describes how to configure a rule for packet matching based on the RTSP.

4.10.1.1 Configure Match Conditions

This section describes how to configure match conditions for a term in an RTSP rule.

URI

To configure packet matching based on URI, include the following statement:

```
Ericsson(config)# epg pgw service-identification rtsp-rule
<rule-name> term <term-id> from rtsp uri
   include-uri-handling
      case
   is <string>
   not-is <string>
   starts-with <string>
   not-starts-with <string>
   ends-with <string>
   not-ends-with <string>
   contains <string>
   not-contains <string>
```

Note: The maximum number of match conditions that can be configured is 10.

By default, case is not considered. The optional `case` statement toggles sensitivity to case.

Enable Escape Character Conversion

To enable conversion of escape characters before evaluation of the URI, include the following statement:

```
Ericsson(config)# epg pgw service-set <service-set-name> serv
ice-identification unescape-conversion
   rtsp
```

This configuration is valid for all DPI rules inspecting RTSP traffic in the service set. Conversion of escape characters can also be enabled for all DPI rules in the EPG, see Section 4.15 on page 60.

RTSP Domain

To configure packet matching based on RTSP domain, include the following statement:

```
Ericsson(config)# epg pgw service-identification rtsp-rule
<rule-name> term <term-id> from rtsp domain
   is <string>
   not-is <string>
   starts-with <string>
   not-starts-with <string>
   ends-with <string>
   not-ends-with <string>
   contains <string>
   not-contains <string>
   domain-length <number>
```

Note: The maximum number of match conditions that can be configured is 10.

It is possible to include RTSP domain together with other match conditions in a term. Only one RTSP domain statement is allowed in a term.

Wildcard characters are partially supported for textual matching configuration. For more information on how to use wildcard characters in RTSP domain configuration, refer to Packet Inspection and Service Classification (PISC).

RTSP Response Codes

To configure packet matching based on RTSP response codes, include the following statement:

```
Ericsson(config)# epg pgw service-identification rtsp-rule
<rule-name> term <term-id> from rtsp response-code
   is (<value> | <value>-<value>)
```

Note: Response codes are matched only in RTSP responses, not in requests.

The configurable value range is 100–551. Response codes can be configured with or without event tracking configured in the rule. Response codes cannot be configured together with URI tracking in the rule. Refer to Charging Methods Configuration for more information.

4.10.1.2 Configure Resulting SDF-ID

To configure the SDF-ID to assign to traffic that matches the conditions in a term, include the following statement:

```
Ericsson(config)# epg pgw service-identification rtsp-rule
<rule-name> term <term-id> then
   payload <sdf-id>
   rtp-payload <sdf-id>
   rtcp-payload <sdf-id>
```

RTP and RTCP payload SDF-IDs can be optionally configured in an RTSP rule term if the following conditions are met:

— `enable-rtcp-detection` is configured in the associated service set. See Section 3.6 on page 13 for more information.

— The match condition response code is not configured in an RTSP rule.

When `enable-rtcp-detection` is configured at the service set level, the following applies to the RTP and RTCP payload SDF-IDs:

— If `rtp-payload` and `rtcp-payload` are not configured, all RTSP traffic with or without RTP and RTCP contents is classified in `payload`.

— If `rtp-payload` and `rtcp-payload` are configured, they must not be configured separately. The RTSP traffic without RTP and RTCP contents is classified in `payload`. RTP traffic is classified in `rtp-payload`, and RTCP traffic is classified in `rtcp-payload`.

4.10.1.3 Configure RTSP Implicit Redirection

To enable support for RTSP implicit redirection, include the following statement:

```
Ericsson(config)# epg pgw service-set <name> service-ident
ification enable-implicit-redirection
   rtsp
```

If an RTSP rule is restricted to a set of configured network addresses or prefixes, RTSP implicit redirection can be detected only if the address or the prefix corresponding to the redirected host or hosts are also configured.

4.10.2 Configure RTSP Rule Set

A rule set can contain one or several rules. The rules are evaluated in descending order by a user-defined `rule-id`. The `rule-id` is a numerical value from 1 through 999,999.

To add a rule to a rule set, include the following statement:

```
Ericsson(config)# epg pgw service-identification rtsp-
rule-set <name> rule <rule-id>
   name <rule-name>
```

4.10.3 Associate RTSP Rule Set with Header Rule

To associate a rule set with a header rule, include the following statement:

```
Ericsson(config)# epg pgw service-identification header-rule
<rule-name> term <term-id> then protocol-inspection
   rtsp-rule-set <name>
```

4.11 Configure SIP Inspection

This section describes configuration of SIP inspection, see also Section 4 on page 14.

4.11.1 Configure SIP Rule

This section describes how to configure a rule for packet matching based on the SIP.

4.11.1.1 Configure Match Conditions

This section describes how to configure match conditions for a term in a SIP rule.

Request URI

To configure packet matching based on Request URI, include the following statement:

```
Ericsson(config)# epg pgw service-identification sip-rule
<rule-name> term <term-id> from sip request-uri
   include-uri-handling
      case
   is <string>
   not-is <string>
   starts-with <string>
   not-starts-with <string>
   ends-with <string>
   not-ends-with <string>
   contains <string>
   not-contains <string>
```

Note: The maximum number of match conditions that can be configured is 10.

By default, case is not considered. The optional `case` statement toggles sensitivity to case.

Enable Escape Character Conversion

To enable conversion of escape characters before evaluation of the URI, include the following statement:

```
Ericsson(config)# epg pgw service-set <service-set-name> serv
ice-identification unescape-conversion
   sip
```

This configuration is valid for all DPI rules inspecting SIP traffic in the service set. Conversion of escape characters can also be enabled for all DPI rules in the EPG, see Section 4.15 on page 60.

VIA Header

To configure packet matching based on the contents of the VIA header, include the following statement:

```
Ericsson(config)# epg pgw service-identification sip-rule
<rule-name> term <term-id> from sip via
   include-uri-handling
      case
   starts-with <string>
   not-starts-with <string>
   ends-with <string>
   not-ends-with <string>
   contains <string>
   not-contains <string>
```

Note: The maximum number of match conditions that can be configured is 10.

By default, case is not considered. The optional `case` statement toggles sensitivity to case. When a SIP packet contains several VIA headers, the EPG concatenates them with a pipe character (|) between the strings before matching the value against the configured string.

Request Type

To configure packet matching based on a request type, include the following statement:

```
Ericsson(config)# epg pgw service-identification sip-rule
<rule-name> term <term-id> from sip
   request-type (invite | ack | bye | cancel | register
| options | info | message | publish | refer | update
| prack | subscribe | notify)
```

Note: Only one request type can be chosen per configuration.

Response Code

To configure packet matching based on a response code or a range of response codes, include the following statement:

```
Ericsson(config)# epg pgw service-identification sip-rule
<rule-name> term <term-id> from sip response-code
   is (<value> | <value>-<value>)
```

The configurable range of response codes is 100–699.

4.11.1.2 Configure Resulting SDF-ID

To configure the SDF-ID to assign to traffic that matches the conditions in a term, include the following statement:

```
Ericsson(config)# epg pgw service-identification sip-rule
<rule-name> term <term-id> then
   payload <sdf-id>
   rtp-payload <sdf-id>
   rtcp-payload <sdf-id>
```

Example 16 shows an example of configuring a payload SDF-ID in a SIP rule term.

```
epg pgw service-identification rtsp-rule rtsp-rs1-r1
 term 1
  name rtsp-rs1-r1-t1
  from rtsp
  then payload 1200
 !
!
epg pgw service-identification rtsp-rule-set rtsp-rs1
 rule 1
  name rtsp-rs1-r1
 !
!
epg pgw service-identification sip-rule sip-rs1-r1
 term 1
  name sip-rs1-r1-t1
  then payload 1100
  from sip
 !
!
epg pgw service-identification sip-rule-set sip-rs1
 rule 1
  name sip-rs1-r1
 !
!
epg pgw service-identification header-rule hdr-rs1-r1
 term 1
  name hdr-rs1-r1-t1
  then protocol-inspection sip-rule-set sip-rs1
  from network-port [ 5060 ]
 !
 term 2
  name hdr-rs1-r1-t2
  then protocol-inspection rtsp-rule-set rtsp-rs1
  from network-address [ 10.170.0.5 ]
 !
!
```

Example 16 Configuring a Payload SDF-ID in a SIP Rule Term

RTP and RTCP payload SDF-IDs can be optionally configured in a SIP rule term if the following conditions are met:

— `enable-rtcp-detection` is configured in the associated service set. See Section 3.6 on page 13 for more information.

— The match condition VIA header or response code is not configured in a SIP rule.

When `enable-rtcp-detection` is configured at the service set level, the following applies to the RTP and RTCP payload SDF-IDs:

— If `rtp-payload` and `rtcp-payload` are not configured, all SIP traffic with or without RTP and RTCP contents is classified in `payload`.

— If `rtp-payload` and `rtcp-payload` are configured, they must not be configured separately. The SIP traffic without RTP and RTCP contents is classified in `payload`. RTP traffic is classified in `rtp-payload`, and RTCP traffic is classified in `rtcp-payload`.

4.11.2 Configure SIP Rule Set

A rule set can contain one or several rules. The rules are evaluated in descending order by a user-defined `rule-id`. The `rule-id` is a numerical value from 1 through 999,999.

To add a rule to a rule set, include the following statement:

```
Ericsson(config)# epg pgw service-identification sip-r
ule-set <name> rule <rule-id>
   name <rule-name>
```

4.11.3 Associate SIP Rule Set with Header Rule

To associate a rule set with a header rule, include the following statement:

```
Ericsson(config)# epg pgw service-identification header-rule
<rule-name> term <term-id> then protocol-inspection
   sip-rule-set <name>
```

4.11.4 Configure Packet Matching Based on Extended SIP Headers

To configure packet matching based on extended SIP headers, do the following:

1. Configure an extended header.

2. Configure a SIP rule using the extended header.

The following sections describe steps 1 and 2. A complete example is given at the end of the section.

Note: A maximum of four extended SIP headers can be configured per service set. All the extended SIP headers must have a textual content.

4.11.4.1 Configure Extended Headers

Extended headers can be configured on the service identification level. The header-name must be composed of printable ASCII characters (from 33 through 126).

Note: Configuring extended SIP headers increases resource consumption and can have an impact on the capacity and performance of the EPG.

To configure packet matching based on an extended SIP header with textual content, include the following statement:

```
Ericsson(config)# epg pgw service-identification extended-header-definition sip
   text-content <header-name>
```

4.11.4.2 Configure SIP Rule Using Extended Headers

To configure a rule to match the textual content of an extended SIP header, include the following statement:

```
Ericsson(config)# epg pgw service-identification sip-r
ule <rule-name> term <term-id> from sip extended-head
er text-content <header-name>
   case
   is <string>
   not-is <string>
   starts-with <string>
   not-starts-with <string>
   ends-with <string>
   not-ends-with <string>
   contains <string>
   not-contains <string>
```

Note: The maximum number of match conditions that can be configured is 10.

4.12 Configure SMTP Inspection

This section describes the configuration of Simple Mail Transfer Protocol (SMTP) inspection, see also Section 4 on page 14.

4.12.1 Configure SMTP Rule

This section describes how to configure a rule for packet matching based on the SMTP.

4.12.1.1 Configure Match Conditions

This section describes how to configure match conditions for a term in an SMTP rule.

Packets are implicitly matched to the SMTP `DATA` command, but not to other commands.

Sender

Configure packet matching of the DATA command, based on the sender attribute in the MAIL command, by including the following statement:

```
Ericsson(config)# epg pgw service-identification smtp-rule
<rule-name> term <term-id> from smtp sender
   case
   is <string>
   not-is <string>
   starts-with <string>
   not-starts-with <string>
   ends-with <string>
   not-ends-with <string>
   contains <string>
   not-contains <string>
```

Note: The maximum number of match conditions that can be configured is 10.

By default, case is not considered. The optional case statement toggles sensitivity to case.

This matches the SMTP DATA command including both requests and responses.

SMTP Operations

Configure packet matching based on SMTP MAIL command by including the following statement:

```
Ericsson(config)# epg pgw service-identification smtp-rule
<rule-name> term <term-id> from smtp operation
   mail
```

This matches the SMTP MAIL command including both requests and responses.

Sender and SMTP Operations

Configuring both the 'Sender' and the operation 'Mail' in the same term matches the Sender in the MAIL command and counts both requests and responses belonging to the MAIL command.

4.12.1.2 Configure Resulting SDF-ID

Configure the SDF-ID to assign to traffic that matches the conditions in a term by including the following statement:

```
Ericsson(config)# epg pgw service-identification smtp-rule
<rule-name> term <term-id> then
   payload <sdf-id>
```

4.12.2 Configure SMTP Rule Set

A rule set can contain one or several rules. The rules are evaluated in descending order by a user-defined `rule-id`. The `rule-id` is a numerical value from 1 through 999,999.

Add a rule to a rule set by including the following statement:

```
Ericsson(config)# epg pgw service-identification smtp-rule-set
<rule-set-name> rule <rule-id>
   name <rule-name>
```

4.12.3 Associate SMTP Rule Set with Header Rule

Associate a rule set with a header rule by including the following statement:

```
Ericsson(config)# epg pgw service-identification header-rule
<rule-name> term <term-id> then protocol-inspection
   smtp-rule-set <name>
```

4.12.4 Configure SMTP Content Enrichment

For information how to configure content enrichment, refer to Content Enrichment Configuration.

4.13 Configure TFTP Inspection

This section describes configuration of Trivial File Transfer Protocol (TFTP) inspection, see also Section 4 on page 14.

4.13.1 Configure TFTP Rule

This section describes how to configure a rule for packet matching based on the TFTP.

4.13.1.1 Configure Match Conditions

This section describes how to configure match conditions for a term in a TFTP rule.

Filename

To configure packet matching based on filename, include the following statement:

```
Ericsson(config)# epg pgw service-identification tftp-rule
<rule-name> term <term-id> from tftp filename
   case
   is <string>
```

```
   not-is <string>
   starts-with <string>
   not-starts-with <string>
   ends-with <string>
   not-ends-with <string>
   contains <string>
   not-contains <string>
```

Note: The maximum number of match conditions that can be configured is 10.

By default, case is not considered. The optional `case` statement toggles sensitivity to case.

TFTP Operations

To configure packet matching based on TFTP operations, include one or several of the following statements:

```
Ericsson(config)# epg pgw service-identification tftp-rule
<rule-name> term <term-id> from tftp
   operation (read-request | write-request)
```

4.13.1.2 Configure Resulting SDF-ID

To configure the SDF-ID to assign to traffic that matches the conditions in a term, include the following statement:

```
Ericsson(config)# epg pgw service-identification tftp-rule
<rule-name> term <term-id> then
   payload <sdf-id>
```

4.13.2 Configure TFTP Rule Set

A rule set can contain one or several rules. The rules are evaluated in descending order by a user-defined `rule-id`. The `rule-id` is a numerical value from 1 through 999,999.

To add a rule to a rule set, include the following statement:

```
Ericsson(config)# epg pgw service-identification tftp-
rule-set <name> rule <rule-id>
   name <rule-name>
```

4.13.3 Associate TFTP Rule Set with Header Rule

To associate a rule set with a header rule, include the following statement:

```
Ericsson(config)# epg pgw service-identification header-rule
<rule-name> term <term-id> then protocol-inspection
```

```
   tftp-rule-set <name>
```

4.14 Configure SSL/TLS Inspection

This section describes configuration of Transport Layer Security (TLS) and Secure Sockets Layer (SSL), see also Section 4 on page 14.

4.14.1 Configure SSL/TLS Rule

This section describes how to configure a rule for packet matching based on the SSL/TLS.

4.14.1.1 Configure Match Conditions

This section describes how to configure match conditions for a term in an SSL/TLS rule.

Note: Strings configured in the following match conditions are not case-sensitive.

Classification of TLS Midflows

To configure packet matching based on TLS midflows, include the following statement:

```
Ericsson(config)# epg pgw service-identification ssl-tls-rule
<rule-name> term <term-id> from
   mid-flow
```

Classification of Segmented Client Hello Messages

To enable the classification of segmented Client Hello messages, include the following statement:

```
Ericsson(config)# epg pgw service-identification ssl-tls-rule
<rule-name> term <term-id> from client-hello
   is-segmented
```

Classification of Unsegmented Client Hello Messages

To enable the classification of unsegmented Client Hello messages, include the following statement:

```
Ericsson(config)# epg pgw service-identification ssl-tls-rule
<rule-name> term <term-id> from client-hello
   not-is-segmented
```

Classification of Out-of-Order Client Hello Segments

To configure packet matching for out-of-order Client Hello uplink packets, include the following statement:

```
Ericsson(config)# epg pgw service-identification ssl-tls-rule
<rule-name> term <term-id> from client-hello
   uplink-out-of-order
```

SNI

To configure packet matching based on the SNI field of the Client Hello handshake message, include the following statement:

```
Ericsson(config)# epg pgw service-identification ssl-tls-rule <rul
e-name> term <term-id> from client-hello server-name-indication
   is <string>
   not-is <string>
   starts-with <string>
   not-starts-with <string>
   ends-with <string>
   not-ends-with <string>
   contains <string>
   not-contains <string>
```

Note: The maximum number of match conditions that can be configured is 10.

Extension Number

To configure packet matching based on the Extension Number field of the Client Hello handshake message, include the following statement:

```
Ericsson(config)# epg pgw service-identification ssl-tls-rule
<rule-name> term <term-id> from client-hello extensions
   contains <string>
```

Note: The maximum number of match conditions that can be configured is 10.

Subject Common Name

To configure packet matching based on the Subject Common Name field of the Certificate handshake message, include the following statement:

```
Ericsson(config)# epg pgw service-identification ssl-tls-rule
<rule-name> term <term-id> from server-certificate common-name
   is <string>
   not-is <string>
   starts-with <string>
   not-starts-with <string>
   ends-with <string>
   not-ends-with <string>
   contains <string>
```

```
   not-contains <string>
```

Note: The maximum number of match conditions that can be configured is 10.

Organization Name

To configure packet matching based on the Organization Name field of the Certificate handshake message, include the following statement:

```
Ericsson(config)# epg pgw service-identification ssl-tls-rule <rul
e-name> term <term-id> from server-certificate organization-name
   is <string>
   not-is <string>
   starts-with <string>
   not-starts-with <string>
   ends-with <string>
   not-ends-with <string>
   contains <string>
   not-contains <string>
```

Note: The maximum number of match conditions that can be configured is 10.

Subject Alternative Name

To configure packet matching based on the Subject Alternative Name field of the Certificate handshake message, include the following statement:

```
Ericsson(config)# epg pgw service-identification ssl-t
ls-rule <rule-name> term <term-id> from server-certifi
cate subject-alternative-name
   is <string>
   not-is <string>
   starts-with <string>
   not-starts-with <string>
   ends-with <string>
   not-ends-with <string>
   contains <string>
   not-contains <string>
```

Note: The maximum number of match conditions that can be configured is 10.

Application Layer Protocol Negotiation

To configure packet matching based on the Application Layer Protocol Negotiation field of the Server Hello handshake message, include the following statement:

```
Ericsson(config)# epg pgw service-identification
ssl-tls-rule <rule-name> term <term-id> from server-hello
application-layer-protocol
   is <string>
   not-is <string>
   starts-with <string>
   not-starts-with <string>
```

```
   ends-with <string>
   not-ends-with <string>
   contains <string>
   not-contains <string>
```

Note: The maximum number of match conditions that can be configured is 10.

Catch-All SSL/TLS

To catch all SSL/TLS traffic not matching any configured field value of the handshake messages, or if no valid server certificate is sent in the SSL/TLS handshake sequence, include the following statement:

```
Ericsson(config)# epg pgw service-identification ssl-tls-rule
<rule-name> term <term-id> from
   any-ssl-tls
```

If the DPI SSL/TLS catch-all rule is explicitly configured, its precedence is higher than heuristic rules. In other words, SSL/TLS masquerading is disabled in this case.

4.14.1.2 Configure Resulting SDF-ID

To configure the SDF-ID to assign to traffic that matches the conditions in a term, include the following statement:

```
Ericsson(config)# epg pgw service-identification ssl-tls-rule
<rule-name> term <term-id> then
   payload <sdf-id>
```

4.14.2 Configure SSL/TLS Rule Set

A rule set can contain one or several rules. The rules are evaluated in descending order by a user-defined `rule-id`. The `rule-id` is a numerical value from 1 through 999,999.

To add a rule to a rule set, include the following statement:

```
Ericsson(config)# epg pgw service-identification ssl-tl
s-rule-set <name> rule <rule-id>
   name <rule-name>
```

4.14.3 Associate SSL/TLS Rule Set with Header Rule

To associate a rule set with a header rule, include the following statement:

```
Ericsson(config)# epg pgw service-identification header-rule
<rule-name> term <term-id> then protocol-inspection
   ssl-tls-rule-set <name>
```

4.14.4 Configure Port Numbers for Application Messages in SSL/TLS Packets

To classify application messages, with protocols such as HTTPS, FTPS, and POPS, carried within SSL/TLS packets, configure the specific protocol name or port number for the application protocol in the header rule associated with the SSL/TLS rule:

Table 1 Supported Port Configuration Values for SSL/TLS Classification

Protocol Name	Port Number
https	443
ftpes	21
ftps	990
ftps-data	989
imaps	993
pops	995
smtps	465

Note: Configuring a header rule for any of the protocols listed in Table 1 prevents that traffic from reaching heuristic rules.

```
Ericsson(config)# epg pgw service-identification header
-rule <rule-name> term <term-id>
   name <term-name>
   from
      network-port <protocol-name/port-number>
   then
      service-data-flow-id
         payload <sdf-id>
      protocol-inspection
         ssl-tls-rule-set <name>
```

If the DPI SSL/TLS catch-all rule is not explicitly configured, the encrypted traffic that does not fall into specific SSL/TLS rules or heuristic rules is classified into the fallback SDF-ID in the associated header rule.

4.14.5 Configure TLS Content Enrichment

For information how to configure content enrichment, refer to Content Enrichment Configuration.

4.15 Enable Escape Character Conversion

To enable conversion of escape characters in HTTP, WSP, RTSP, and SIP URIs for all DPI rules in the EPG, include the following statement:

```
Ericsson(config)# epg pgw service-identification
   unescape-conversion
```

Conversion of escape characters can also be configured per protocol on a service set level. See the sections describing configuration of the respective protocols.

For information on how to enable escape character conversion, see Section 4.6.1.1 on page 22. For more information on escape character conversion, refer to Packet Inspection and Service Classification (PISC).

4.16 Enable Classification of Signalling Packets

In a service set, classification of TCP or WSP setup signalling, TCP or WSP teardown signalling, or any combination of these can be enabled in two different ways:

— Global classification of signalling packets: all matching signalling packets within the service set are classified within the same configured SDF-ID.

— Network IP address aware classification of signalling packets: signalling packets within the service set that match the network IP address and prefix conditions are classified in its own specific SDF-ID.

It is possible to concurrently configure both possibilities, for example, to isolate the signalling toward a pool of servers within a subnet from the rest of signalling traffic. If so, network IP address aware signalling classification has precedence on global classification signalling rules.

For more information on signalling packet classification, refer to Packet Inspection and Service Classification (PISC).

4.16.1 Configure General Signalling Classification

To enable global classification of TCP or WSP setup signalling, TCP or WSP teardown signalling, separate classification of TCP teardown signalling without payload, or any combination of these, include the following statement:

```
Ericsson(config)# epg pgw service-set <service-set-name> serv
ice-identification signaling-classification
   setup
      tcp <sdf-id>
      wsp <sdf-id>
   teardown
      tcp-no-payload <sdf-id>
      tcp <sdf-id>
      wsp <sdf-id>
```

Note: Classification of global teardown signalling is resource consuming and can have an impact on the capacity and performance of the EPG. Teardown configured in a header rule used in several service sets can negatively affect the performance of all service sets.

It is highly recommended to use network address-based teardown signalling classification for a limited subset of IP addresses whenever it is possible.

Classification of setup signalling packets applies to the whole service set. Classification of teardown signalling packets applies only to traffic being classified by a DPI rule.

The EPG can be configured to classify TCP teardown signalling with and without payload:

— Configure the `tcp-no-payload` attribute for TCP teardown traffic without payload only for signalling traffic. This traffic is free of charge.

— Configure the `tcp` attribute for TCP teardown traffic with payload to avoid possible fraud. This traffic can be charged or throttled.

Setting up signalling classification for WSP requires at least one header rule and one HTTP-WSP rule set for WSP to be inspected. The service set for which WSP setup signalling is configured must point to a header rule that includes the `from protocol udp` statement. This header rule must refer to an HTTP-WSP rule set, using the `then protocol-inspection http-wsp-rule-set set-name` statement. At least one specific HTTP-WSP rule must be present in the HTTP-WSP rule set that matches WSP traffic.

Global signalling rules for TCP require at least one additional header rule pointing to an HTTP-WSP rule set. However, network address-based signalling rules that fall within the address range of the additional header rule have a higher priority than the global signalling rules.

4.16.1.1 Configure Strict TCP Teardown Signalling

To enable packets related to payload (either ACK or pure payload packet) received when TCP teardown is started to be classified as payload, include the following statement:

```
Ericsson(config)# epg pgw service-set <service-set-name> serv
ice-identification signaling-classification teardown
   tcp-strict-teardown
```

Note: To activate `tcp-strict-teardown`, `tcp <sdf-id>` must be configured.

4.16.2 Configure Classification Based on Network Address

To enable classification of TCP or WSP setup or teardown signalling for a header rule depending on the targeted network address, include the following statement:

Note: Classification of TCP or WSP teardown signalling is resource consuming and can have an impact on the capacity and performance of the EPG.

```
Ericsson(config)# epg pgw service-identification header
-rule <rule-name> term <term-id>
   name <term-name>
   from
      network-prefix <network-prefix>
      network-address <network-address>
      protocol
         tcp
            signaling
               setup
               teardown
         udp
            wsp
               signaling
                  setup
                  teardown
   then
      service-data-flow-id
         payload <sdf-id>
```

Note: The classification of network IP address-based teardown signalling is resource consuming and may have an impact on the capacity and performance of the EPG. Keep the number of teardown signalling `network-address` entries as low as possible.

The following restrictions apply to network address-based configuration:

— The signalling header rules based on network address for WSP require at least one additional header rule pointing to an HTTP-WSP rule set. The network address range specified in the `From` clause in the additional header rule must include the network address specified in the signalling header rule. Network address-based signalling rules have precedence over the rules in the rule set of the additional header rule, and global signalling rules.

— The configuration of a complete header rule is rejected if at least one `network-address` or `network-prefix` and also at least one `setup` or `teardown` parameter are not present.

— The configuration of TCP or WSP signalling classification is not allowed in the same header rule term as DPI (`protocol-inspection`) or traffic redirection (`redirect-unauthorized`).

— A maximum of 10 entries of `network-address` or `network-prefix` can be configured per header rule.

— Port-based and domain-based signalling rules cannot be configured.

```
epg pgw service-identification header-rule h1
 term 1
  name t1
  then service-data-flow-id payload 40
  from network-prefix [ 10.0.0.1/8 ]
  from network-address [ 10.0.0.0 ]
  from protocol tcp signaling setup
  from protocol udp wsp signaling setup
 !
!
```

Example 17 Classification Based on Network Address

4.17 Configure Flow Time out per Service Set

To configure the flow time-out per service set, include the following statement:

```
Ericsson(config)# epg pgw service-set <service-set-name
> service-identification flow-timeout
   (dns | ftp | http | imap | mms | pop3 | rtp | rtsp | sip | smtp | tcp | tftp
      timeout <seconds>
```

The configured values for time-out are expressed in seconds and have a value range of 1–3600. If not configured, the following default values are used:

— DNS, HTTP, IMAP, MMS, POP3, SMTP, TCP, TFTP, TLS, UDP, WSP, WTP: 30

— RTP: 60

— FTP, RTSP, SIP: 300

The configured service set name must exist in the EPG configuration. For more information, refer to SACC Configuration.

Note: The flow time-out configuration has an impact on the EPG memory and capacity. Do not change the default time-out settings unless there are capacity problems that make it advisable to reduce the default settings or well-known service aware stateful inspection problems (for example, related to content enrichment) that make it advisable to increase the default settings. Contact Ericsson support before changing the default flow time-out settings.

Special attention is needed when configuring the flow time-out for the following protocols:

HTTP

The default time-out value for HTTP flows is 30 seconds. However, if HTTP flows are content enriched, it may be necessary to increase the time-out for the HTTP flows. The time-out for the TCP flows must be set to the same value as that for the HTTP flows that are content enriched. Otherwise, after a time-out,

packets belonging to the same TCP flow may be classified into different SIs, and thus may be charged differently. Contact Ericsson support for more information on such cases.

TCP

After the first TCP SYN packet is received on an unestablished TCP connection, the normal flow timer is set to six seconds, which means that in this state the GGSN or PGW terminates the flow after six seconds of inactivity, regardless of the configured flow time-out.

After the first TCP FIN packet is received, the normal flow timer is set again to six seconds, which means that in this state the GGSN or PGW terminates the flow after six seconds of inactivity, regardless of the configured flow time-out.

Note: When the GGSN or PGW terminates the flow after six seconds of inactivity, certain packets can be classified to unexpected SDF-IDs.

RTP

The default value for RTP streams handled in the RTSP is 60 seconds, which is the default value for pause time-out in RTSP. If the streaming server is configured to support longer time-outs, then the timer must be increased to the same value.

4.18 Configure TCP Deferred Charging

The following steps describe how to configure TCP deferred charging in the EPG:

1. Enable TCP deferred charging on service set level.

   ```
   Ericsson(config)# epg pgw service-set <name>
   Ericsson(config-service-set-<name>)# service-identification
   enable-tcp-setup-deferred-charging
   ```

2. Configure the default SDF-ID for TCP deferred charging on service set level.

   ```
   Ericsson(config-service-set-<name>)# service-identification
   service-data-flow-id default tcp-setup-signaling <sdf-id>
   ```

3. By default, TCP deferred charging is disabled for all services. Optionally, TCP deferred charging can be enabled or disabled for all or individual services, using either of the following options:

   a Enable TCP deferred charging for all services and disable TCP deferred charging for individual services, identified by an SDF-ID, or a consecutive range of SDF-IDs.

   ```
   Ericsson(config)# epg pgw rule-space <name>
   Ericsson(config-rule-space-<name>)# deferred-charging
   tcp-setup default enabled
   ```

```
Ericsson(config-rule-space-<name>)# deferred-charging
tcp-setup disable-service-data-flow-id (<sdf-id> |
<sdf-id>-<sdf-id>)
```

b Disable TCP deferred charging for all services and enable TCP deferred charging for individual services, identified by an SDF-ID, or a consecutive range of SDF-IDs.

```
Ericsson(config)# epg pgw rule-space <name>
Ericsson(config-rule-space-<name>)# deferred-charging
tcp-setup default disabled
Ericsson(config-rule-space-<name>)# deferred-charging
tcp-setup enable-service-data-flow-id (<sdf-id> |
<sdf-id>-<sdf-id>)
```

For detailed information on TCP deferred charging and supported protocols, refer to Packet Inspection and Service Classification (PISC).

4.19 Enable HTTP/2 Inspection

To enable inspection of HTTP/2 traffic for the EPG, include the following statement:

```
Ericsson(config)# epg pgw service-identification
   enable-http2-inspection
```

By default, the HTTP/2 inspection is disabled. For more information on HTTP/2, refer to Packet Inspection and Service Classification (PISC).

4.20 Tethering Detection

TTL-based tethering detection can be activated at two levels:

— Service set level

— Header rule level

However, it cannot be configured at both levels simultaneously.

For TTL-based tethering detection, the EPG calculates the hop count value based on the default TTL value or values. To configure default TTL values for TTL-based tethering detection, include the following statement:

```
Ericsson(config)# epg pgw service-identification tethering-settings
   ttl-default-values <ttl-value>
```

Attribute	Explanation
`ttl-default-values`	The supported values of this attribute are 1–255. A maximum of 32 default values can be configured. By default, the following values are used for the hop count calculation: 1, 32, 64, 128, and 255. If the `ttl-default-values` attribute is configured, the configured value or values overwrite the default values, except the 255 value, which is hard-coded. The same default value cannot be repeated in the configuration.

Note: Default TTL values can be configured without enabling TTL-based tethering detection at service set level or at header rule level.

Example 18 shows how to configure default TTL values for tethering detection.

```
Ericsson(config)# epg pgw service-identification tethering-settings ttl-defa
```

Example 18 Default TTL Values for TTL-Based Tethering Detection

4.20.1 Enable TTL-Based Tethering Detection at Service Set Level

To activate TTL-based tethering detection at service set level and configure the default payload used for all detected tethering traffic, include the following statement:

```
Ericsson(config)# epg pgw service-set <service-set-na
me> service-identification
   tethering-default-payload <sdf-id>
```

Including the `tethering-default-payload` parameter activates TTL-based tethering detection for all rules defined in a service set. The `tethering-default-payload` value is returned to the EPG when tethering traffic is detected in the UE.

If the tethering device is not detected, traffic is classified according to the legacy rules configured in the service set.

For further information on tethering detection, refer to Packet Inspection and Service Classification (PISC).

4.20.2 Enable TTL-Based Tethering Detection at Header Rule Level

To activate TTL-based tethering detection at header rule level and configure the payload to assign the tethering traffic, include the following statement:

```
Ericsson(config)# epg pgw service-identification header
-rule <rule-name> term <term-id>
```

```
   from
      traffic-type tethering
   then
      service-data-flow-id
         payload <sdf-id>
```

Optionally, TTL-based tethering detection at header rule level can be activated with user agent filtering in the DPI rule. For more information about how to configure user agent filtering, refer to Traffic Redirection Configuration.

A tethering rule is created based on the configured classification conditions:

— Traffic Type Tethering. For an example configuration, see Example 19.

— Traffic Type Tethering and User Agent Filtering. For an example configuration, see Example 20.

```
epg pgw service-set ss-gx-1
 service-identification service-data-flow-id default payload 999
 service-identification header-rule-sets ruleset3
 !
!
epg pgw service-identification header-rule rule3
 term 1
  then service-data-flow-id payload 111
  from traffic-type tethering
  from protocol tcp
 !
!
epg pgw service-identification header-rule-set ruleset3
 rule 1
  name rule3
 !
!
```

Example 19 Enabling TTL-Based Tethering Detection at Header Rule Level - Traffic Type Tethering

```
epg pgw service-set ss-gx-1
 service-identification service-data-flow-id default payload 999
 service-identification header-rule-sets ruleset3
 !
!
epg pgw service-identification http-wsp-rule http_r1
 term 1
  then payload 666
  from http extended-header text-content User-Agent
   contains [ EPG ]
  !
 !
!
epg pgw service-identification http-wsp-rule-set http_rs1
 rule 1
  name http_r1
 !
!
epg pgw service-identification header-rule rule3
 term 1
  then service-data-flow-id payload 111
  then protocol-inspection http-wsp-rule-set http_rs1
  from traffic-type tethering
  from protocol tcp
 !
!
epg pgw service-identification header-rule-set ruleset3
 rule 1
  name rule3
 !
!
```

Example 20 Enabling TTL-Based Tethering Detection at Header Rule Level - Traffic Type Tethering and User Agent Filtering

If the tethering device is detected and the traffic is classified in a tethering rule, the payload configured in this rule is associated with this traffic. If the tethering device is detected, but no tethering rule is associated with this traffic, the DPI analysis continues. If the tethering device is not detected, the traffic is not classified in this rule.

Tethering detection defined at header rule level applies the same policy of priorities as the other rules. Therefore, to prioritize tethering rules, define them first.

A default catch-all rule can also be configured for all tethering traffic that does not match any tethering rule configured. Depending on the rules priority, this catch-all rule can be used to block all tethering traffic that does not match a service configured with the tethering rules defined.

TTL-based tethering detection at header rule level is compatible with signalling classification. However, TTL-based tethering detection at header rule level

with user agent filtering is not compatible with signalling classification. See the restriction in Section 4.16.2 on page 62.

```
epg pgw service-identification header-rule rule3
 term 1
  then service-data-flow-id payload 111
  from traffic-type tethering
  from network-address [ 216.58.201.130 ]
  from protocol tcp signaling setup
  from protocol tcp signaling teardown
 !
 term 2
  then service-data-flow-id payload 112
  from network-address [ 216.58.201.130 ]
  from protocol tcp signaling setup
  from protocol tcp signaling teardown
 !
!
epg pgw service-identification header-rule-set ruleset3
 rule 1
  name rule3
 !
!
```

Example 21 Enabling TTL-Based Tethering Detection at Header Rule Level - Traffic Type Tethering and TCP Signalling Classification

4.21 Enable TTL Identification at Service Set Level

To activate TTL identification at service set level, include the following statement:

```
Ericsson(config)# epg pgw service-set <service-set-name> serv
ice-identification ttl-identification
   ttl-lower-than <ttl-value>
   ttl-payload <sdf-id>
```

Attribute	Explanation
`ttl-lower-than`	Traffic with a TTL that is lower than this value is classified in the configured SDF-ID.
`ttl-payload`	The SDF-ID where traffic that matches the `ttl-lower-than` attribute is classified.

Note: This configuration is valid for IPv4 and IPv6, even though the TTL field has been renamed hop limit in IPv6.

Example 22 shows that any traffic that has TTL equal to or lower than 2 is classified into the SDF-ID 34.

```
Ericsson(config)# epg pgw service-set ss1
Ericsson(config-service-set-ss1)# service-identification ttl-identification
Ericsson(config-service-set-ss1)# service-identification ttl-identification
```

Example 22 Enabling TTL Identification at Service Set Level

For more information on TTL identification at service set level, refer to Packet Inspection and Service Classification (PISC).

4.22 Enable TTL Identification at Header Rule Level

To configure TTL match conditions in a header rule, include the following statement:

```
Ericsson(config)# epg pgw service-identification header-rule
<rule-name> term <term-id> from ttl-uplink <ttl-uplink-id>
   is <string>
   not-is <string>
   lower-than <string>
   higher-than <string>
```

Attribute	Explanation
`is`	Uplink packets with this TTL value match the condition. The supported values of this attribute are 1–255.
`not-is`	Uplink packets with a different TTL value match the condition. The supported values of this attribute are 1–255.
`lower-than`	Uplink packets with a lower TTL value match the condition. The supported values of this attribute are 2–256.
`higher-than`	Uplink packets with this TTL value match the condition. The supported values of this attribute are 0–254.

The following configuration rules apply:

— The maximum number of `ttl-uplink` classes that can be configured is 10.

— The maximum number of match conditions that can be configured in a `ttl-uplink` class is 10. If more than one condition is configured, the following rules apply:

- `is` condition cannot be configured with the rest of the conditions.
- Only one of `is`, `lower-than`, and `higher-than` condition can be configured.
- More than one `not-is` condition can be configured.

— A DPI rule can be associated with a header rule where TTL identification is configured.

— The EPG supports TTL identification at header rule level for IPv4 and IPv6, even though the TTL field has been renamed hop limit in IPv6.

Example 23 shows that any traffic that has TTL equal to 60 in the first uplink packet of the connection is classified into the SDF-ID with payload 5.

```
epg pgw service-identification header-rule hr1
 term 1
  then service-data-flow-id payload 5
  from ttl-uplink 1
   is 60
  !
 !
!
```

Example 23 Enabling TTL Identification at Header Rule Level

Example 24 shows that any traffic that has TTL different than 64 in the first uplink packet of the connection and an HTTP URI containing `www.example.com` is classified into the SDF-ID with payload 8.

```
epg pgw service-identification http-wsp-rule hwr1
 term 1
  name t1
  then payload 8
  from uri contains [ www.example.com ]
 !
!
epg pgw service-identification http-wsp-rule-set hwrs1
 rule 1
  name hwr1
 !
!
epg pgw service-identification header-rule hr2
 term 1
  then protocol-inspection http-wsp-rule-set hwrs1
  from ttl-uplink 1
   not-is [ 64 ]
  !
 !
!
```

Example 24 Enabling TTL Identification at Header Rule Level Associated with DPI Rule

For more information on TTL identification at header rule level, refer to Packet Inspection and Service Classification (PISC).

5 Configure Heuristic Packet Inspection

To configure heuristic packet inspection, the following actions are mandatory:

— Enable a heuristics package.

— Configure a heuristic rule.

— Configure a heuristic rule set.

— Associate the heuristic rule set with a service set.

5.1 Update Protocols Supported by Heuristic Packet Inspection

Update the protocols supported by heuristic packet inspection by installing a new heuristics package. A new heuristics package can be installed in two ways:

— By upgrading or updating the EPG software

— By manually installing a heuristics package during runtime

To update during runtime, download the latest heuristics package from the Ericsson Software Gateway, at https://swgateway.ericsson.net.

Note: To be able to download a heuristics package, an account with appropriate privileges on the e-business portal is required.

To install a heuristics package, include the following statement:

```
Ericsson(config)# epg pgw service-identification
   install-heuristics-package package <path/heuristics-package>
```

Note: If the EPG has two Node Management Boards (NMBs), entering the `install-heuristics-package` command on either the active NMB or the standby NMB installs the heuristics package on both NMBs.

To enable a new heuristics package, include the following statement:

```
Ericsson(config)# epg pgw service-identification
   heuristics-package <heuristics-package>
```

To display version information about the currently installed heuristics package, and a full list of supported protocols, include the following statement:

```
Ericsson(config)# epg pgw service-identification
   heuristics-information
```

The printout from the command lists all protocols supported by the heuristics package, and states the category to which they belong. Deprecated protocols are also listed and identified. Example 25 shows a partial example printout from the command.

```
heuristics-package:
    name: Heur2014b
    version: 4.3142

    heuristics-protocols:

        heuristics-protocol:
            name: bit-torrent
            category: file-transfer

        heuristics-protocol:
            name: edonkey
            category: file-transfer

        heuristics-protocol:
            name: gnutella
            category: file-transfer

        heuristics-protocol:
            name: skype
            category: voip

        heuristics-protocol:
            name: yahoo-messenger
            status: deprecated
            category: none
```

Example 25 Heuristics Package Printout Example

To display information about a specific heuristics package that has been downloaded, include the following statement:

```
Ericsson(config)# epg pgw service-identification
   heuristics-information-package package <heuristics-package>
```

To remove a heuristics package from both NMBs, include the following statement:

```
Ericsson(config)# epg pgw service-identification
   uninstall-heuristics-package package <heuristics-package>
```

5.2 Configure Heuristic Rule

A heuristic rule consists of one or several terms. The terms are evaluated in descending order by a user-defined `term-id`. The `term-id` is a numerical value from 1 to 999,999. There is no difference between configuring several rules with

one term each, or one rule with several terms. The EPG capacity is only affected by the total number of terms, refer to EPG Characteristics.

To configure a term in a heuristic rule, the following actions are mandatory:

— Configure the match conditions for the protocols to be classified.

  If several match conditions are configured in a term, all conditions must be fulfilled for the term to match. Multiple match conditions in the same term are handled in a logical `AND` relation.

— Configure the resulting SDF-ID.

5.2.1 Configure Match Condition

A complete list of protocols and categories of protocols supported by heuristic packet inspection is available in Heuristics Supported Protocols and Compatibility, and in the release notes contained in the heuristics package.

Support for protocols can be configured individually or by category. For more information on heuristic protocol configuration, refer to Heuristic Configuration Guidelines.

To configure heuristic inspection for one or more protocols, include the following statement:

```
Ericsson(config)# epg pgw service-identification heuristic-rule
<rule-name> term <term-id> from
   protocol <protocol-name>
```

To configure heuristic inspection for a category of protocols, include the following statement:

```
Ericsson(config)# epg pgw service-identification heuristic-rule
<rule-name> term <term-id> from
   category <category-name>
```

Note: Configuring a category and a protocol belonging to that category results in a nondeterministic classification, with traffic being classified in either the SDF-ID assigned to the protocol or in the SDF-ID assigned to the category, if both SDF-IDs are different. To prevent such a behavior, it is recommended to configure either only specific protocols in the category, or only the category. Configuring all the protocols in the category is equivalent to configuring the category.

5.2.2 Configure Resulting SDF-ID

To configure the SDF-ID assigned to traffic that matches a term, include the following statement:

```
Ericsson(config)# epg pgw service-identification heuristic-rule
<rule-name> term <term-id> then
   payload <sdf-id>
```

5.3 Configure Heuristic Rule Set

A heuristic rule set can contain one or several heuristic rules. The rules are evaluated in ascending order by a user-defined `rule-id`. The `rule-id` is a numerical value from 1 through 999,999.

To add a rule to a rule set, include the following statement:

```
Ericsson(config)# epg pgw service-identification heurist
ic-rule-set <name> rule <rule-id>
   name <heuristic-rule-name>
```

5.4 Associate Heuristic Rule Set with Service Set

One or more heuristic rule sets can be associated with a service set.

Note: A service set must not contain more than one rule for each protocol. This avoids incorrect classification of the inspected traffic.

To associate one or more heuristic rule sets with a service set, include the following statement:

```
Ericsson(config)# epg pgw service-set <service-set-na
me> service-identification
   heuristic-rule-sets <rule-set-name>
      priority <rule-set-priority>
```

Note: The `priority` attribute is used to sort the configured rule sets list, so that the rule sets are chosen according to the ordered list.

5.5 Configure HTTP Masquerading Detection for Service Set

If HTTP masquerading detection is enabled, packets matching a header rule associated with an HTTP-WSP rule set but not matching any HTTP-WSP rules are evaluated against the heuristic rules configured for the service set. HTTP masquerading detection is disabled by default.

To enable HTTP masquerading detection for a service set, include the following statement:

```
Ericsson(config)# epg pgw service-set <service-set-name> serv
ice-identification enable-masquerading-detection
   http
```

HTTP masquerading detection can only be enabled for service sets containing heuristic rule sets.

5.5.1 Enable URI-Based Redirection for HTTP Default Classification with HTTP Masquerading

To enable URI-based redirection for HTTP packets that match the default SDF-ID in the service set and if configured, a header rule fallback SDF-ID, include the following statement:

```
Ericsson(config)# epg pgw service-set <service-set-name> serv
ice-identification enable-masquerading-detection
   http-enable-redirect
```

Note: URI-based redirection for HTTP default classification is enabled only if HTTP masquerading detection is enabled.

For information about `httpEnableRedirect`, refer to Traffic Redirection Configuration.

5.5.2 Activate URI Tracking for HTTP Default Classification with HTTP Masquerading

To activate URI tracking for HTTP packets that match the default SDF-ID in the service set and if configured, a header rule fallback SDF-ID, include the following statement:

```
Ericsson(config)# epg pgw service-set <service-set-name> serv
ice-identification enable-masquerading-detection
   http-enable-implicit-uri-tracking
```

Note:

- URI tracking for HTTP default classification is activated only if HTTP masquerading detection is enabled.
- URI tracking for HTTP default classification uses a high percentage of node resources.

6 Security

This section describes the PISC configurations for security.

6.1 Fraud Prevention

The EPG provides the following mechanisms against fraud:

— Fraud can occur when there is a charging configuration service that is zero-rated. If the service is defined too broadly, without any restrictions, hackers can tunnel through the service and navigate freely to the internet. The solution against fraud is to define zero-rated services as narrowly as needed, and to add restrictions to the command definition. For more information, refer to Packet Inspection and Service Classification (PISC).

— The EPG validates HTTP formatting to detect malformed packets. Malformed packets can exploit known errors in proxy implementations and allow unintentional free browsing. For more information, see Section 6.1.1 on page 78.

— The EPG can be configured to classify segmented or unsegmented HTTP and TLS traffic that comes after flows time out (midflows) because this traffic can be subject to fraud. For more information, see Section 4.6.1.2 on page 24 and Section 4.14.1.1 on page 56.

— The EPG can be configured to classify out-of-order HTTP and TLS uplink traffic because this traffic can be subject to fraud. For more information, see Section 4.6.1.2 on page 24 and Section 4.14.1.1 on page 56.

— The EPG can be configured to classify TCP teardown traffic with payload because this traffic can be subject to fraud. For more information, see Section 4.16.1 on page 61.

6.1.1 Detect Wrong HTTP Format

The EPG prevents possible fraud by using validation check to detect wrong HTTP formatting. For more information, refer to Packet Inspection and Service Classification (PISC).

To classify non-compliant HTTP messages into an SDF-ID, include the following statements:

```
Ericsson(config)# epg pgw service-set <service-set-name> serv
ice-identification security http1x-wrong-format
   service-data-flow <service-identifier>
```

When a packet is classified in a fraud SDF-ID, all subsequent packets in the flow are classified into the same SDF-ID, until the next HTTP request comes.

6.2 Limitation of Sessions or Flows

To limit the maximum number of simultaneous sessions per user when SACC functionality is enabled, include the following statements:

```
Ericsson(config)# epg pgw service-identification flow-limits
   max-user-child <number>
   max-node-flow <number>
   max-analyzer-routing <number>
```

The configurable parameters are described as follows:

— `max-user-child`

  Maximum number of allowed flows per protocol and user. The minimum value is `0`, which means no specific limit of the number of flows per protocol and user. The default value is `500`.

  This setting defines the maximum number of concurrent active Deep Packet Inspection (DPI) analysis flows in a user IP session, permitted per analyzer. This setting can be used to limit the maximum number of simultaneous sessions and to protect the user plane vSFO from possible SYN or port-scanning attacks without degrading the user plane vSFO capacity.

  When the maximum number of concurrent active DPI analysis flows in a user IP session is exceeded for an analyzer, no more new flows are created for such an analyzer, traffic is best effort classified and is forwarded.

  Note: The `max-user-child` value may need to be increased in certain cases such as header redirect, content enrichment, and Peer-to-Peer (P2P). For example, P2P traffic can generate a high number of flows. If P2P traffic filters are configured, it is recommended to set the `max-user-child` value to `1000` or more. However, this also means that the allowed flows for other protocols are increased when P2P is not used. The risk must be considered when changing the `max-user-child` setting.

— `max-node-flow`

  Maximum number of accepted flows in the User Plane vSFO. The value range is `0`–`30,000,000`, where `0` means no specific limit of the number of flows in the User Plane vSFO. The default value depends on the number of DPI threads and the memory available.

  This setting defines the maximum number of concurrent active DPI analysis flows permitted per User Plane vSFO.

  For each User Plane vSFO, the `max-node-flow` value is distributed evenly among all the vCPUs in the User Plane vSFO handling DPI threads.

— `max-analyzer-routing`

  Maximum number of accepted dynamic routing rules per analyzer. The value range is `0`–`5000`, where `0` means no specific limit of the number of dynamic routing rules per analyzer. The default value is `1000`. This setting provides protection against malicious or malformed packets.

A dynamic routing rule is applicable for protocols, such as FTP, RTSP, and SIP, that dynamically negotiate payload connections that are used during an application session. If dynamic routing rules are used for RTSP and SIP protocols to map RTP traffic (user data) with RTSP or SIP traffic (control data) per user, this parameter limits the number of RTP routing rules for RTSP or SIP per DPI instance.

Classifying traffic into a configurable SDF-ID is possible when the maximum number of flows per protocol and user or per user plane vSFO is exceeded. Authorization on the SDF-ID used for traffic that exceeds the upper limit is handled through mapping to an appropriate ACR, and the traffic is either blocked or authorized depending on what is configured for the ACR.

Note: Traffic redirection is not possible by using this function.

To configure an SDF-ID for traffic that exceeds the upper limit, include one of the following statements:

```
Ericsson(config)# epg pgw service-set <service-set-na
me> service-identification
   maximum-user-flows-exceeded <service-identifier>
   maximum-node-flows-exceeded <service-identifier>
```

If configured, the maximum number of user or node flows at service set level has the priority according to the priority order list in Packet Inspection and Service Classification (PISC).

To display statistics regarding the number of active flows, execute the following command:

```
[local]Ericsson# epg pgw
   statistics of pisc-flows
```

For more information on the output of the command, refer to Action Commands for the GGSN and PGW.

Note: It is required that the parameter name of the service set exists in the EPG configuration under the `service-identification` hierarchy level. For details on how to configure a service set, refer to SACC Configuration.

6.3 DDoS Protection

DDoS protection is a license-controlled feature.

DDoS protection can be enabled as follows:

1. A valid license key must be used.
2. Activate the DDoS protection feature.
3. Configure DDoS protection for TCP SYN flood attacks from UEs.

6.3.1 Use a Valid License Key

For more information about license keys, refer to Software License Management.

6.3.2 Activate the DDoS Protection Feature

To activate the DDoS protection feature, include the following statement:

```
Ericsson(config)# epg pgw feature-activation
   ddos-protection
```

6.3.3 Configure DDoS Protection for TCP SYN Flood Attacks from UEs

To configure DDoS protection to detect and classify TCP SYN flood attacks from UEs, include the following statement:

```
Ericsson(config)# epg pgw service-set <service-set-name> serv
ice-identification security tcp-syn-flood
   threshold-detect-uplink <number-of-tcp-handshakes>
   threshold-attach-uplink <number-of-attack-tcp-handshakes>
   threshold-uplink-exceeded-flow-id <service-identifier>
```

— `threshold-detect-uplink` is the detection threshold for ongoing uplink initiated TCP handshakes per user. By default, this threshold is set to 40, and the supported configurable range is 1–65535. Ericsson recommends configuring this threshold according to the following considerations:

  - The threshold must be sufficiently low that it is less than the number of the active flows of a TCP SYN flood attack.
  - The threshold must be sufficiently high that it is exceeded only by a small fraction of legitimate traffic. Typical legitimate traffic that can exceed the threshold is video streaming and web browsing of pages full of advertisements. In the detection state, legitimate traffic is discarded by the detection.

— `threshold-attack-uplink` is the attack threshold for ongoing uplink initiated TCP handshakes per user and per IP destination. By default, this threshold is set to 25, and the supported configurable range is 1–65535. For legitimate traffic, the number of ongoing handshakes toward an IP address is less than the total number of ongoing uplink initiated handshakes. For attack traffic, the number of ongoing handshakes toward an IP address is high. This threshold can differentiate between legitimate and attack traffic.

— `threshold-uplink-exceeded-flow-id` is the SDF-ID where the uplink attack packets are classified. This is a mandatory parameter because attack mitigation is based on service classification and policy enforcement.

It is not allowed to configure `threshold-detect-uplink` or `threshold-attack-uplink` to a value greater than or equal to the value configured for

`max-user-child`, the maximum allowed flows per protocol and user. For more information on `max-user-child`, see Section 6.2 on page 78.

7 Optimization

The impact of the PISC configuration described earlier in the document on the capacity and performance of the EPG can be mitigated by optimizing the SACC-related configuration.

For detailed information and recommendations that help optimize the SACC-related configuration, refer to SACC Optimization.

Reference List

Standards

[1] Format for Literal IPv6 Addresses in URL's, RFC 2732, RFC 2732

[2] IP Version 6 Addressing Architecture, RFC 2373, RFC 2373

[3] Uniform Resource Identifiers (URI): Generic Syntax, RFC 2396, RFC 2396

[4] Uniform Resource Identifier (URI): Generic Syntax, RFC 3986, RFC 3986

Online Documentation

[5] IANA: Assigned Internet Protocol Numbers, http://www.iana.org/assignments/protocol-numbers