



## **Cisco ASR 5000 Series Enhanced Charging Services Administration Guide Addendum Version 12.2**

**Last Updated November 30, 2011**

### **Americas Headquarters**

Cisco Systems, Inc.  
170 West Tasman Drive  
San Jose, CA 95134-1706  
USA  
<http://www.cisco.com>  
Tel: 408 526-4000  
800 553-NETS (6387)  
Fax: 408 527-0883

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The following information is for FCC compliance of Class A devices: This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to part 15 of the FCC rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio-frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference, in which case users will be required to correct the interference at their own expense.

The following information is for FCC compliance of Class B devices: This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If the equipment causes interference to radio or television reception, which can be determined by turning the equipment off and on, users are encouraged to try to correct the interference by using one or more of the following measures:

Reorient or relocate the receiving antenna.

Increase the separation between the equipment and receiver.

Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.

Consult the dealer or an experienced radio/TV technician for help.

Modifications to this product not authorized by Cisco could void the FCC approval and negate your authority to operate the product.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Cisco and the Cisco Logo are trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and other countries. A listing of Cisco's trademarks can be found at [www.cisco.com/go/trademarks](http://www.cisco.com/go/trademarks). Third party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

Cisco ASR 5000 Series Enhanced Charging Services Administration Guide Addendum

© 2011 Cisco Systems, Inc. and/or its affiliated entities. All rights reserved.

# CONTENTS

---

<b>About this Guide .....</b>	<b>v</b>
Conventions Used.....	vi
Contacting Customer Support .....	vii
<b>Affected Documents .....</b>	<b>9</b>
<b>DNS Snooping .....</b>	<b>11</b>
DNS Snooping Feature Overview .....	12
Licensing .....	12
Bulkstatistics Support .....	13
How it Works.....	14
Limitations and Dependencies.....	18
DNS Snooping Feature Configuration .....	20
CLI Command Reference.....	21
ACS Configuration Mode Commands.....	21
ip dns-learnt-entries .....	21
ACS Ruledef Configuration Mode Commands .....	23
ip server-domain-name .....	23
Exec Mode Commands.....	24
clear active-charging dns-learnt-ip-addresses .....	24
show active-charging dns-learnt-ip-addresses .....	25
Global Configuration Mode Commands.....	27
threshold dns-learnt-ip-max-entries .....	27
<b>Tethering Detection .....</b>	<b>29</b>
Tethering Detection Feature Overview .....	30
MUR Support for Tethering Detection .....	31
Tethering Detection Databases .....	32
OS Signature Database .....	32
UA Signature Database.....	33
TAC Database.....	33
Loading and Upgrading Tethering Detection Databases .....	34
Session Recovery Support .....	34
Limitations and Dependencies.....	34
Tethering Detection Feature Configuration.....	35
Upgrading Tethering Detection Databases .....	35
Sample Configurations .....	36
CLI Command Reference.....	41
ACS Configuration Mode Commands.....	41
tethering-database .....	41
ACS Rulebase Configuration Mode Commands .....	43
tethering-detection .....	43
ACS Ruledef Configuration Mode Commands .....	44
tethering-detection .....	44
EDR Format Configuration Mode Commands .....	46
rule-variable .....	46
Exec Mode Commands.....	48
clear active-charging tethering-detection statistics .....	48
show active-charging tethering-detection .....	49

upgrade tethering-detection .....	51
-----------------------------------	----





# About this Guide

---

This document pertains to the features and functionality that run on and/or that are related to the Cisco® ASR 5000 Chassis, formerly the Starent Networks ST40.

## Conventions Used

The following tables describe the conventions used throughout this documentation.

Icon	Notice Type	Description
	Information Note	Provides information about important features or instructions.
	Caution	Alerts you of potential damage to a program, device, or system.
	Warning	Alerts you of potential personal injury or fatality. May also alert you of potential electrical hazards.
	Electro-Static Discharge (ESD)	Alerts you to take proper grounding precautions before handling a product.

Typeface Conventions	Description
Text represented as a <code>screen display</code>	This typeface represents displays that appear on your terminal screen, for example: <code>Login:</code>
Text represented as <b>commands</b>	This typeface represents commands that you enter, for example: <b><code>show ip access-list</code></b> This document always gives the full form of a command in lowercase letters. Commands are not case sensitive.
Text represented as a <b>command variable</b>	This typeface represents a variable that is part of a command, for example: <b><code>show card slot_number</code></b> <code>slot_number</code> is a variable representing the desired chassis slot number.
Text represented as menu or sub-menu names	This typeface represents menus and sub-menus that you access within a software application, for example: Click the <b>File</b> menu, then click <b>New</b>

Command Syntax Conventions	Description
{ <b>keyword</b> or <i>variable</i> }	Required keywords and variables are surrounded by grouped brackets. Required keywords and variables are those components that are required to be entered as part of the command syntax.
[ <b>keyword</b> or <i>variable</i> ]	Optional keywords or variables, or those that a user may or may not choose to use, are surrounded by square brackets.
	With some commands there may be a group of variables from which the user chooses one. These are called alternative variables and are documented by separating each variable with a vertical bar (also known as a pipe filter). Pipe filters can be used in conjunction with required or optional keywords or variables. For example: <code>{ <b>nonce</b>   <b>timestamp</b> }</code> OR <code>[ <b>count</b> <i>number_of_packets</i>   <b>size</b> <i>number_of_bytes</i> ]</code>

## Contacting Customer Support

Use the information in this section to contact customer support.

**For New Customers:** Refer to the support area of <http://www.cisco.com> for up-to-date product documentation or to submit a service request. A valid username and password is required to this site. Please contact your local sales or service representative for additional information.

**For Existing Customers with support contracts through Starent Networks:** Refer to the support area of <https://support.starentnetworks.com/> for up-to-date product documentation or to submit a service request. A valid username and password is required to this site. Please contact your local sales or service representative for additional information.



**Important:** For warranty and repair information, please be sure to include the Return Material Authorization (RMA) tracking number on the outside of the package.





# Chapter 1

## Affected Documents

---

This addendum provides new and/or expanded information pertaining to the Enhanced Charging Services documentation delivered as part of the 12.2 release.

Documentation updates provided in this addendum pertain to the documents listed in the following table and correspond to the stated release date(s):

Document	Part Number	Release Date
<i>Cisco ASR 5000 Series Enhanced Charging Services Administration Guide: Version 12.2</i>	OL-25591-01	October 17, 2011
<i>Cisco ASR 5000 Series Command Line Interface Reference: Version 12.2</i>	OL-25551-01	October 17, 2011



# Chapter 2

## DNS Snooping

---

This chapter describes the DNS Snooping feature.

This chapter covers the following topics:

- [DNS Snooping Feature Overview](#)
- [DNS Snooping Feature Configuration](#)
- [CLI Command Reference](#)

# DNS Snooping Feature Overview

This section provides an overview of the DNS Snooping feature.



**Important:** In the 12.2 release, the DNS Snooping feature is supported only on the GGSN and P-GW.

ECS, using L7 rules, can be configured to filter subscriber traffic based on domain name. While this works fine for HTTP-based traffic, a subscriber's initial HTTP request may result in additional flows being established that use protocols other than HTTP and/or may be encrypted. Also, a domain may be served by multiple servers, each with its own IP address. This means that using an IP rule instead of an HTTP rule will result in multiple IP rules, one for each server “behind” the domain. This necessitates service providers to maintain a list of IP addresses for domain-based filters.

The DNS Snooping feature enables a set of IP rules to be installed based on the response from a DNS query. The rule in this case contains a fully qualified domain name (for example, m.google.com) or its segment (for example, google) and a switch that causes the domain to be resolved to a set of IP addresses. The rules installed are thus IP rules. Any actions specified in the domain rule are inherited by the resulting IP rules.

When configured, DNS snooping will be done on live traffic for every subscriber.

The DNS Snooping feature enables operators to create ruledefs specifying domain names or their segments. On defining the ruledefs, the gateway will monitor all the DNS responses sent towards the UE, and will snoop only the DNS response that has q-name or a-name as specified in the rules, and identify all the IP addresses resulting from the DNS response. A table of these IP addresses is maintained per destination context per rulebase per instance and shared across subscribers of the same destination context same rulebase per instance. In case DNS queries made by different subscribers produce different results, all the IP entries in the table are stored based on their Time to Live (TTL) and the configurable timer. The TTL or the timer whichever is greater is used for aging out the IP entry. Dynamic IP rules are created for these IP entries within the same rule having the domain name, applying the same charging action to these dynamic rules. This solution will have the exact IP entries as obtained live from snooping DNS responses. They will be geographically and TTL correct.

## Licensing

DNS Snooping is a licensed feature.



**Important:** Use of DNS Snooping feature requires that a valid license key be installed on the ASR chassis. Contact your local Sales or Support representative for information on how to obtain a license.

## Bulkstatistics Support

Bulkstatistics reporting for the DNS Snooping feature is supported.

In the ECS schema, the following new bulkstatistics are available in support of the DNS Snooping feature:

- **ecs-dns-learnt-ipv4-entries:** The total number of learnt IPv4 entries.  
Increments if a new IPv4 entry is received, and decrements if the entry gets timed out and gets flushed, or when the rule line corresponding to an IPv4 entry is removed from the rulebase.  
Type: Counter  
Date Type: Int64
- **ecs-dns-flushed-ipv4-entries:** The total number of flushed IPv4 entries.  
Increments if the TTL for an IPv4 entry expires. When the rule lines (URLs to be snooped) are removed from the rulebase, the counter is set to 0.  
Type: Counter  
Date Type: Int64
- **ecs-dns-replaced-ipv4-entries:** The total number of replaced IPv4 entries.  
Increments if the TTL value of that entry is replaced with a new value. If the rule lines (URLs to be snooped) are removed from the rulebase, the counter is set to 0.  
Type: Counter  
Date Type: Int64
- **ecs-dns-overflown-ipv4-entries:** The total number of overflown IPv4 entries.  
Increments if the number of learnt DNS entries exceeds “ACS maximum learnt IPv4 entries per pool” or “ACS maximum learnt IPv4 entries across system”. If the rule lines (URLs to be snooped) are removed from the rulebase, the counter is set to 0.  
Type: Counter  
Date Type: Int64
- **ecs-dns-learnt-ipv6-entries:** The total number of learnt IPv6 entries.  
Increments if a new IPv6 entry is received, and decrements if the entry gets timed out and gets flushed, or when the rule line corresponding to an IPv6 entry is removed from the rulebase.  
Type: Counter  
Date Type: Int64
- **ecs-dns-flushed-ipv6-entries:** The total number of flushed IPv6 entries.  
Increments if the TTL for an IPv6 entry expires. When the rule lines (URLs to be snooped) are removed from the rulebase, the counter is set to 0.  
Type: Counter  
Date Type: Int64

- **ecs-dns-replaced-ipv6-entries:** The total number of replaced IPv6 entries.  
 Increments if the TTL value of that entry is replaced with a new value. When the rule lines (URLs to be snooped) are removed from the rulebase, the counter is set to 0.  
 Type: Counter  
 Date Type: Int64
- **ecs-dns-overflown-ipv6-entries:** The total number of overflown IPv6 entries.  
 Increments if the number of learnt DNS entries exceeds “ACS maximum learnt IPv6 entries per pool” or “ACS maximum learnt IPv6 entries across system”. If the rule lines (URLs to be snooped) are removed from the rulebase, the counter is set to 0.  
 Type: Counter  
 Date Type: Int64

## How it Works

This section describes how the DNS Snooping feature works.

ECS allows operators to create ruledefs specifying domain names or their segments using options available in the CLI ruledef syntax (contains, starts-with, ends with, or equal to). This allows operators to match all the traffic going to specified fully qualified domain names as presented by the UE in the DNS queries, or segments of the domain names.

Internally, when a ruledef containing ip server-domain-name keyword is defined and the ruledef is used in a rulebase, an IP table, similar to the following, is created per rulebase per instance.

Operator	Domain Name	IP Pool Pointer	Associated Ruledef	List of CNAMEs
contains	gmail	ip-pool1	domain_google	l.google.com
=	yahoo.com	ip-pool2	domain_yahoo	
starts-with	gmail	ip-pool3	domain_start_gmail	

On definition of the ruledefs, the gateway will monitor all the DNS responses sent towards the UE and will snoop the DNS responses from valid DNS servers. IP addresses (IPv4 and IPv6) resulting from the DNS responses are learnt dynamically and will be used for further rule matching. These dynamic Service Data Flows (SDFs), containing IP addresses, may also be reused by ECS for other subscribers from the same routing instance in order to classify the subscriber traffic.

The dynamic SDFs generated are kept for the TTL specified in the DNS response plus a configurable timer that can be added to the TTL in case the DNS response contains a very small TTL.



**Important:** If the rule created using this feature is removed from the configuration then all the associated dynamic SDFs are removed immediately. The usage incurred by the subscriber for traffic matching the removed SDFs will be reported over the Gy interface when the usage reporting for the corresponding rating group is due.

In case DNS queries made by different subscribers produce different results, all the dynamically generated SDFs are stored based on their TTL and the configured timer.

DNS Snooping supports DNS responses containing nested CNAME responses.

When the DNS response contains nested CNAME record, a list per entry in the IP-table is dynamically allocated to store the CNAMEs. CNAME is the canonical name of the alias, which means the q-name to which the actual query was made is the alias name and this CNAME is the actual domain name to which the query should be made. So, the IP addresses found in response to CNAME DNS query is stored in the same IP-pool as that of the alias.

Here, either the DNS response to the actual alias contains CNAME record along with its A record or only the CNAME record. In the first case the IP address is already resolved for CNAME and it is included in the learnt IP addresses IP-pool.

In both the scenarios, the list of CNAMEs is stored in the same record of the IP-table, which is keyed by operator+domain. By default, the operator for CNAME is "equal". So, while snooping DNS responses, DNS responses for a-name as in the CNAME list will also be snooped and the IP addresses stored in the corresponding IP-pool. This allows the feature to work in case DNS responses have nested CNAME response.

Like IP addresses, even CNAME entries have TTL associated with them. In the same five minute timer, where the aged IP addresses are timed out, the CNAME entries will also be looked at and the expired CNAME entries reference removed from the corresponding entry.

The DNS Snooping feature supports both IPv4 and IPv6 addresses. The maximum limits being:

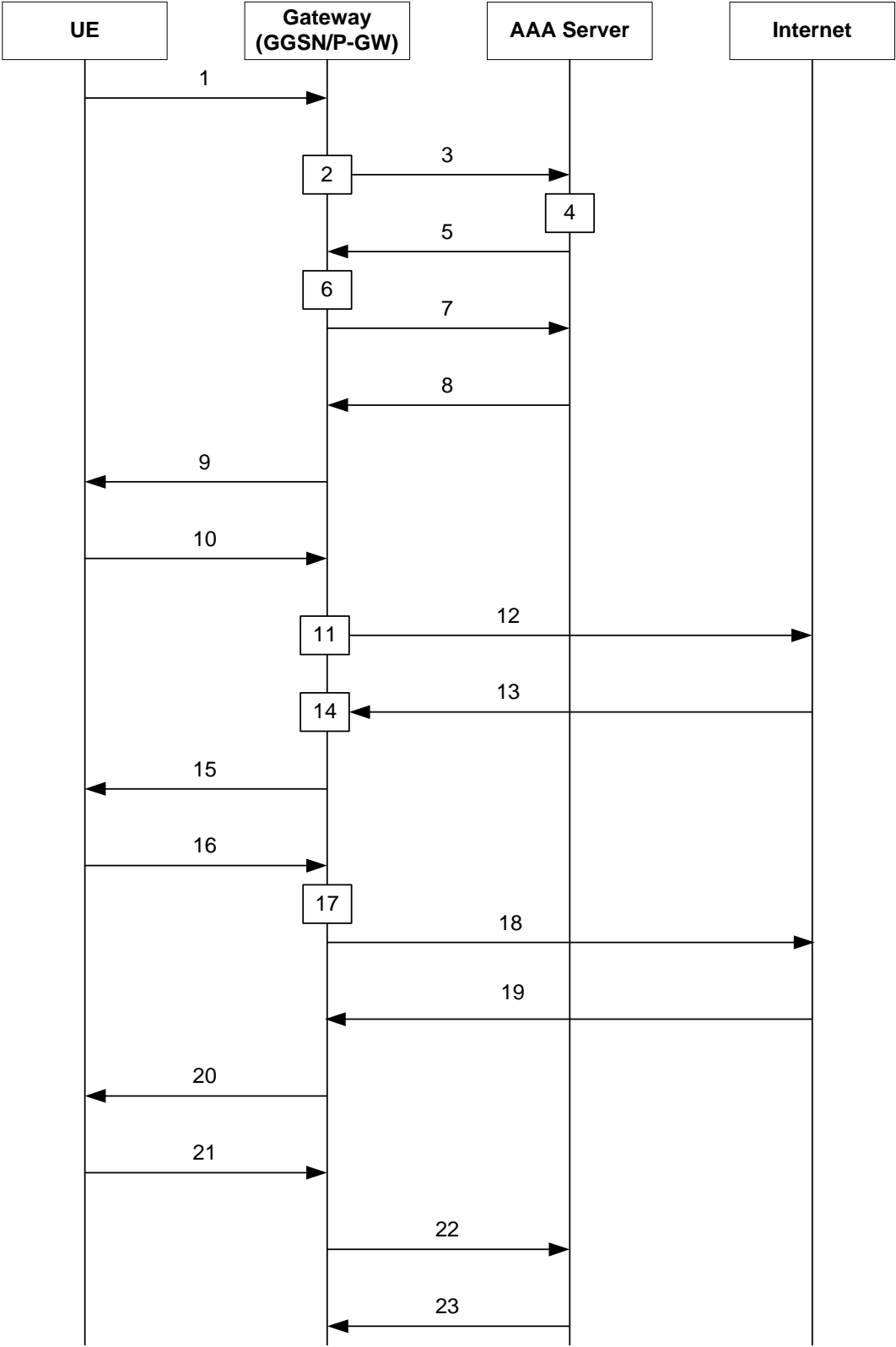
- IPv4 addresses learnt per server-domain-name pattern: 200
- IPv4 addresses learnt per instance across all IPv4 pools: 51200
- IPv6 addresses learnt per server-domain-name pattern: 100
- IPv6 addresses learnt per instance across all IPv6 pools: 25600

Rule matching: While matching rule for IP packets, it will be checked if the source IP address matches any of the entries stored in the IP pools formed as part of DNS snooping. If a match is found, the corresponding ruledef is determined from the IP table. The other rule lines of the rule are matched, and if it is the highest priority rule matched it is returned as a match. The corresponding charging-action is applied. So the same priority as that of the domain name is applied to its corresponding IP addresses, and is matched as a logical OR of the domain or the IP addresses.

Lookup (matching) is performed in learnt IP pools only for the first packet of the ADS as the destination IP address will not change for that flow, and will match the same rule (last rule matched for this ADS flow) for all the packets of the flow. This enables to have the same rule matched even if its IP addresses get aged out when the flow is ongoing.

The following call flow illustration and descriptions describe how the DNS Snooping feature works.

Figure 1. DNS Snooping Call Flow





**Table 1. DNS Snooping Call Flow Descriptions**

Step No.	Description
1	UE requests for registration to the system.
2	System processes UE-related information with ECS subsystem.
3	System sends AAA Access Request to AAA server for UE.
4	The AAA server processes the AAA Access Request from the ECS to create the session, and the Policy Manager in AAA server uses subscriber identification parameters including NAI (username@domain), Calling Station ID (IMSI, MSID), and Framed IP Address (HoA) as the basis for subscriber lookup.
5	<p>The Policy Manager and AAA generate and send an Access Accept message including all policy and other attributes to establish the session to ECS.</p> <p>The Policy Manager and/or AAA include following attributes in the Access Accept message:</p> <ul style="list-style-type: none"> <li>Filter ID or Access Control List Name: Applied to subscriber session. It typically contains the name of the Content Service Steering (CSS) ACL. The CSS ACL establishes the particular service treatments such as Content Filtering, ECS, Traffic Performance Optimization, Stateful Firewall, VPN, etc. to apply to a subscriber session and the service order sequence to use in the inbound or outbound directions. Real-time or delay sensitive flows are directly transmitted to the Internet with no further processing required. In this case, no CSS ACL or Filter ID is included in the Access Response.</li> <li>SN1-Rulebase Name: This custom attribute contains information such as consumer, business name, child/adult/teen, etc.). The rulebase name identifies the particular rule definitions to apply. Rulebase definitions are used in ECS as the basis for deriving charging actions such as prepaid/postpaid volume/duration/destination billing and charging data files (EDRs/UDRs). Rulebase definitions are generated in the Active Charging Configuration Mode and can be applied to individual subscribers, to domains or on per-context basis.</li> </ul>
6	ECS creates a new session for UE, and sends the rulebase to ACS subsystem if required.
7	ECS sends Accounting-Start messages to the AAA server.
8	The AAA server sends Accounting-Start response message to ECS.
9	ECS establishes data flow with UE.
10	UE requests for data with URL name (DNS query).
11	ECS analyzed the query-name from the DNS query of the subscriber, and if it matches the entry in the “DNS URLs to be snooped” list (created when ip server-domain-name rules were defined in rulebase), it marks this request for its response to be snooped.
12	DNS query is sent to the Internet.
13	DNS response is received from the Internet.
14	Based on the various answer records in the response the IP addresses are snooped and included in the “list of learnt IP addresses”.
15	DNS response is sent to the UE.
16	Actual URL request comes from the UE.

Step No.	Description
17	Looking at the server-ip-address of this packet, rule matching will be done based on the “list of learnt IP addresses” and the rules already configured. An action is taken based on the ruledef matched and the charging action configured.
18	If the packet is to be forwarded, it is forwarded to the Internet.
19	A Response is received from the Internet.
20	The response is sent to the UE.
21	UE requests for session termination.
22	System sends Accounting-Stop Request to AAA server.
23	AAA server stops accounting for subscriber and sends Accounting-Stop-Response to the system.

## Limitations and Dependencies

This section identifies limitations and dependencies for the DNS Snooping feature.

- On a SessMgr kill or card switchover, the dynamic IP rules created based on domain name resolution will be lost. Till the time a new DNS query is made, the dynamic IP based rules will not be applied. These rules will be recreated on new DNS traffic. So, SessMgr recovery is not supported for these dynamic IP rules.
- The `ip server-domain-name` ruledef can be used as a predefined dynamic rule, static rule, or as a part of group of ruledefs. However, it cannot be used as a dynamic-only rule as dynamic-only rules apply up to L4 and this is an L7 rule.
- Operators will have to define the valid domain-name servers, the DNS responses from which will be considered correct and snooped and included in the list of dynamic-learnt IP addresses. If the list of valid domain-name servers is not provided, then the DNS responses from all DNS servers will be considered valid and included in the list of learnt-ip-addresses. This can cause an issue of including invalid IP addresses in the list, in the case a subscriber makes DNS query to his own created DNS server and hacks the response being sent. In this case, these IP addresses will be learnt and this traffic may be free-rated or blocked incorrectly depending on the action set. The above config is suggested to avoid any kind of attacks on DNS traffic.
- There will a limit on total number of learnt-ip addresses per server-domain-name ruledef for memory and performance considerations. Any more IP addresses across this limit will not be learnt and hence the charging-action will not be applied to these IP addresses. Similarly there will be a limit on the total number of server-domain-name ruledefs that can be configured.
- If same IP is returned in DNS responses for different DNS q-names (same IP hosting multiple URLs), then while rule matching, the higher priority rule having this learnt-ip address will be matched. This can have undesired rule matching as explained below:

For example, if DNS queries for both `www.facebook.com` and `www.cnn.com` returned IP address `162.168.10.2`. Here we have allow action for domain `www.facebook.com` and block or no action for `www.cnn.com` which is at a lower priority than allow rule. In this if the actual request for `www.cnn.com` comes than as the server IP is same, it will match the higher priority allow rule for domain `www.facebook.com` (considering there are no other rule lines or all lines match) and thus, free rated incorrectly. However, this will happen only of same IP address is returned for different q-names, which is rare and cannot be handled.

- The lookup for IPv6 learnt IP addresses will not be optimized. Hash based lookup (optimization) is done for IPv4 addresses lookup. In a future release Longest Prefixed Match (LPM) based optimization will be considered for both IPv4 and IPv6 learnt IP address matching.
- DNS snooping embedded URLs will not function with HTTP TPO (Traffic Performance Optimization in-line service). HTTP TPO preemptively resolves host names present in embedded URLs of HTML content and rewrites the same with resolved IP addresses. In this case client will not send a DNS query and hence our current implementation will not be able to snoop DNS responses for these embedded URLs. This will be addressed in a future release.

## DNS Snooping Feature Configuration

This section describes how to configure the DNS Snooping feature.

To configure the DNS Snooping feature use the following configuration:

**configure**

```
    active-charging service <ecs_service_name>

        ip dns-learnt-entries timeout <timeout_period>

        ruledef <ruledef_name>

            ip server-domain-name { = | contains | ends-with | starts-with }
            <domain_name/domain_name_segment>

            ...

        exit

    rulebase <rulebase_name>

        action priority <priority> ruledef <ruledef_name> charging-action
        <charging_action_name>

        ...

    end
```

# CLI Command Reference

This section provides details of new/modified CLI commands required to configure the DNS Snooping feature.

## ACS Configuration Mode Commands

### ip dns-learnt-entries

This command configures how long to keep the snooped IPv4 addresses that were extracted from DNS responses.

#### Product

ACS

#### Privilege

Security Administrator, Administrator

#### Syntax

```
ip dns-learnt-entries timeout timeout_period
{ default | no } ip dns-learnt-entries timeout
```

---

#### default

Configures this command with the default DNS-learnt-entries timeout setting.  
Default: 300 seconds

---

#### no

Specifies to always use the TTL value in the DNS response, and not the timeout configured with this command.

---

#### *timeout\_period*

Specifies the DNS-learnt-entries timeout period, in seconds.  
*timeout\_period* must be an integer from 1 through 2147483647.

---

#### Usage

Use this command to configure how long to keep the snooped IPv4 addresses that were extracted from DNS responses—for the TTL specified in the DNS response, or for the time period configured with this command, if greater.

The configurable timer will be at global ECS level and shared across all IP addresses. Internally, a five minute (non configurable) timer will be started whenever DNS analyzer is enabled. On timeout of this timer, all the learnt IP addresses will be checked for TTL expiry and the expired entries will be flushed.

---

**Example**

The following command specifies to keep the snooped IPv4 addresses that were extracted from DNS responses for a time period of *900* seconds, or for the TTL value specified in the DNS response, whichever is greater:

```
ip dns-learnt-entries timeout 900
```

## ACS Ruledef Configuration Mode Commands

### ip server-domain-name

This command defines rule expressions to match host names (domain names).

#### Product

All

#### Privilege

Security Administrator, Administrator

#### Syntax

**[ no ] ip server-domain-name** *operator domain\_name/domain\_name\_segment*

---

**no**

Deletes the specified rule expression.

---

*operator*

Specifies how to logically match the server domain name.

*operator* must be one of the following:

- **=**: Equals
- **contains**: Contains
- **ends-with**: Ends with
- **starts-with**: Starts with

---

*domain\_name/domain\_name\_segment*

Specifies the domain name to match, either the FQDN or a part of it.

*domain\_name/domain\_name\_segment* must be an alpha and/or numeric string of 1 through 127 characters in length.

---

#### Usage

Use this command to define rule expressions to match full or partial host names (domain names). The rule will be matched for the learnt IP addresses resolved from DNS query to the specified domain names. DNS responses for the specified domain names will be snooped and the learnt IP addresses stored. Besides being used for standard rule matching, this command also enables the DNS Snooping feature if the rulebase references any ruledefs with this configuration. The DNS protocol analyzer must also be enabled in the rulebase.

---

#### Example

The following command defines a rule expression to match domain names containing *star*:

```
ip server-domain-name contains star
```

## Exec Mode Commands

### clear active-charging dns-learnt-ip-addresses

This command clears DNS learnt IP address statistics for the DNS Snooping feature.

#### Product

ACS

#### Privilege

Security Administrator, Administrator, Operator

#### Syntax

```
clear active-charging dns-learnt-ip-addresses statistics sessmgr { all |  
instance sessmgr_instance } [ | { grep grep_options | more } ]
```

---

```
sessmgr { all | instance sessmgr_instance }
```

Clears statistics for all or the specified Session Manager (SessMgr) instance.

- **all**: Clears statistics for all SessMgr instances.
- **instance sessmgr\_instance**: Clears statistics for the specified SessMgr instance.  
*sessmgr\_instance* must be an integer from 1 through 65535.

---

```
grep grep_options | more
```

Specifies that the output of this command is to be piped (sent) to the command specified. A command to send the output to must be specified.

---

#### Usage

Use this command to clear DNS learnt IP address statistics for the DNS Snooping feature. On clearing the statistics using this command, only the entries-flushed, entries-replaced, and IP-Overflows statistics are cleared as these are cumulative statistics. Total-entries will not be cleared as it is an instantaneous statistic of the current total entries in that rule line.

---

#### Example

The following command clears all DNS learnt IP address statistics:

```
clear active-charging dns-learnt-ip-addresses statistics sessmgr all
```



## show active-charging dns-learnt-ip-addresses

This command displays DNS learnt IP address statistics for the DNS Snooping feature.

### Product

ACS

### Privilege

Security Administrator, Administrator, Operator, Inspector

### Syntax

```
show active-charging dns-learnt-ip-addresses statistics { sessmgr { all |
instance sessmgr_instance } [ verbose ] | summary } [ | { grep grep_options |
more } ]
```

---

```
sessmgr { all | instance sessmgr_instance } [ verbose ]
```

Displays information for all or the specified Session Manager (SessMgr) instance.

- **all**: Displays information for all SessMgr instances.
- **instance** *sessmgr\_instance*: Displays information for specific SessMgr instance.  
*sessmgr\_instance* must be an integer from 1 through 65535.
- **verbose**: Displays detailed statistics for specified criteria. Use this keyword to view the learnt IP addresses.

---

**summary**

Displays summary information.

---

```
grep grep_options | more
```

Specifies that the output of this command is to be piped (sent) to the command specified. A command to send the output to must be specified.

---

### Usage

Use this command to view statistics for the DNS Snooping feature related DNS learnt-ip-addresses.

This command displays the number of learnt IP entries per rule line. It displays on a service level the number of resolved (learnt) IP addresses per rule line per rulebase (once if a rule line is used multiple times in the same rulebase as it is shared across rulebase) per destination context per SessMgr instance. It also displays the number of entries flushed due to TTL expiry. The field *entries\_replaced* gives the number of entries replaced (same IP returned again) in the pool due to a DNS response by same/another subscriber for same domain-name, wherein the TTL of the entry will be replaced.

IPv4-overflows will start incrementing when the maximum limit of 51200 across system is reached OR limit of 200 per pattern is reached.

Ipv6-overflows will start incrementing when maximum limit of 25600 across system is reached OR limit of 100 per pattern is reached.

Limits are:

- Maximum 51200 IPv4 entries per instance shared across IPv4 all pools.
- Maximum 200 IPv4 entries per pool (pool is same as discussed before (per rule-line pattern)).
- Maximum 25600 IPv6 entries per instance shared across all IPv6 pools.

- Maximum 100 IPv6 entries per pool.

---

**Example**

The following command displays summary statistics for DNS learnt IP addresses:

```
show active-charging dns-learnt-ip-addresses statistics summary
```

## Global Configuration Mode Commands

### threshold dns-learnt-ip-max-entries

This command configures thresholds for the percentage of total DNS-learnt IP entries in relation to the ECS DNS Snooping feature.

#### Product

ECS

#### Privilege

Security Administrator, Administrator

#### Syntax

```
threshold dns-learnt-ip-max-entries high_thresh [ clear low_thresh ]
```

```
default threshold dns-learnt-ip-max-entries
```

---

#### **default**

Configures this command with the default threshold setting.

Default: 90 percent. It is the same for both high and low thresholds.

---

#### *high\_thresh*

Default: 90 percent

The high threshold for percentage of total DNS-learnt IP entries. When the percentage of total DNS-learnt IP entries meets or exceeds the high threshold at the end of the polling interval, an alert or alarm is generated.

When the percentage of total DNS-learnt IPv4 entries meets or exceeds the high threshold, the ECSTotalDNSLearntIPv4Threshold trap is generated.

When the percentage of total DNS-learnt IPv6 entries meets or exceeds the high threshold, the ECSTotalDNSLearntIPv6Threshold trap is generated.

*high\_thresh* must be an integer value from 0 through 100.

When configured to 0 the threshold is disabled.

---

#### **clear** *low\_thresh*

Default: 90 percent

The low threshold for percentage of total DNS-learnt IP entries. When the percentage of total DNS-learnt IP entries goes below the low threshold within the polling interval, a clear alarm is generated.

When the percentage of total DNS-learnt IPv4 entries goes below the low threshold, the ECSTotalDNSLearntIPv4ThresholdClear trap is generated.

When the percentage of total DNS-learnt IPv6 entries goes below the low threshold, the ECSTotalDNSLearntIPv6ThresholdClear trap is generated.

*low\_thresh* must be an integer value from 0 through 100.

When configured to 0 the threshold is disabled.



**Important:** This value is ignored for the Alert model. In addition, if this value is not configured for the Alarm model, the system assumes it is identical to the low threshold.

---

---

**Usage**

Use this command to configure thresholds for the percentage of total DNS-learnt IP entries in relation to the ECS DNS Snooping feature. Note that this threshold applies to both IPv4 and IPv6 DNS entries.

Alerts or alarms are triggered based on the following rules:

- **Enter condition:** Actual percentage of total DNS-learnt IP entries  $\geq$  specified percentage of total DNS-learnt IP entries.
- **Clear condition:** Actual of total DNS-learnt IP entries  $<$  specified clear percentage of total DNS-learnt IP entries.

If a trigger condition exists at the end of the polling interval, the alert or alarm will not be generated until the end of the polling interval.

Refer to the **threshold poll** command to configure the polling interval, and the **threshold monitoring ecs** command to enable thresholding for this value.

---

**Example**

The following command configures a high threshold of 65 percent and a clear threshold of 35 percent for total DNS-learnt IP entries:

```
threshold dns-learnt-ip-max-entries 65 clear 35
```

# Chapter 3

## Tethering Detection

---

This chapter describes the Tethering Detection feature.

This chapter covers the following topics:

- [Tethering Detection Feature Overview](#)
- [Tethering Detection Feature Configuration](#)
- [CLI Command Reference](#)

# Tethering Detection Feature Overview

This section provides an overview of the Tethering Detection feature.



**Important:** In the 12.2 release, the Tethering Detection feature is supported only on the GGSN.

Tethering refers to the use of a mobile smartphone as a USB dongle/modem to provide Internet connectivity to PC devices (laptops, PDAs, tablets, and so on) running on the smartphone's data plan. Typically, for smartphone users, most operators have in place an unlimited data plan, the usage of which is intended to be from the smartphone as a mobile device. However, some subscribers use the low cost / unlimited usage data plan to provide Internet connectivity to their laptops in places where normal Internet connection via broadband/WiFi may be costly, unavailable, or insecure.

The Tethering Detection feature enables detection of subscriber data traffic originating from PC devices tethered to mobile smartphones, and also provides effective reporting to enable service providers take business decisions on how to manage such usage and to bill subscribers accordingly.



**Important:** In the 12.2 release, Tethering Detection is supported only for IPv4 (TCP) traffic flows.

ECS determines tethering detection using a combination of the following client device detection techniques:

- **HTTP UserAgent String based Device Signature Detection**—In this method the HTTP analyzers extract and analyze the UserAgent string from the first HTTP request sent by the MS.  
  
If none of the HTTP requests sent contain the UserAgent string or the UserAgent string sent in the first HTTP request does not match, then the decision is exclusively based on the device's OS fingerprint signature detection.
- **TCP-SYN based Device OS Fingerprint Signature Detection**—In this method the IP (L3) and TCP (L4) analyzers extract and analyze certain values from the following IP and TCP header fields of the first packet of a TCP flow sent by the MS.
  - From IP Header:
    - Overall SYN packet size
    - Initial TTL
    - DF bit
  - From TCP Header:
    - TCP Window size
  - From TCP Options:
    - Maximum Segment Size
    - Window scaling
    - Selective ACK OK
    - Timestamp
    - NOP
    - EOL

- **Mobile Device TAC Number based Detection**—The Type Allocation Code (TAC) number is part of the IMEI number which is available after the call is established. The TAC number of the bearer is looked up in the mobile smartphone TAC database. If a match is found, the actual tethering detection decision for that subscriber session depends on subsequent OS and/or UA match. If required, subsequently ECS performs tethering detection for all flows for that subscriber.



**Important:** Note that TAC number based detection by itself is not a tethering detection method. It only aids in deciding for which of the mobile smartphones connecting to the gateway tethering detection must be carried out. It helps in reducing the scope of tethering detection to only those smartphones that provide users tethering capability.

Since the same smartphone (say iPhone) can concurrently be used as a modem and as a handset, concurrent tethered and non-tethered flows are possible. In this scenario, ECS can detect tethered flows from non-tethered flows. ECS can configure and associate different rating-group/content-id with the usage as a modem vis-à-vis a regular smartphone and be able to do differential charging accordingly for tethered and non-tethered flows.

The Tethering Detection feature is enabled on a per rulebase basis. The rulebase (billing plan) assigned for APN will contain the tethering detection related configuration. ECS performs tethering detection on a per flow basis for all subscribers (for whom TAC database match succeeded) using an APN in which the feature is enabled. The extent to which the detection mechanism is executed depends on the type of flow. If it is a non-TCP flow, for example UDP or ICMP, then tethering detection is not possible for the same.

**Tethering detection on an HTTP flow:** When a subscriber logs onto the service provider network using a mobile smartphone device and performs HTTP transaction from a browser on a tethered device connected to the smartphone, if tethering detection is enabled in the rulebase for the APN used by the subscriber and smartphone TAC is successfully identified, tethering detection will be attempted on the TCP flow of that subscriber.

**Tethering detection on a non-HTTP TCP flow:** When a subscriber logs onto the service provider network using a smartphone device and initiates a TCP connection for a non-HTTP application, such as FTP client or an SNMP mail client, if tethering detection is enabled in the rulebase for the APN used by the subscriber, and smartphone TAC is successfully identified, tethering detection will be attempted on every TCP flow of that subscriber.

## MUR Support for Tethering Detection

The ASR chassis works in conjunction with the Mobility Unified Reporting (MUR) application to facilitate tethering detection on the chassis.

MUR is used to collect samples of HTTP and TCP signatures from live traffic to create a database of OS and UA signatures for assorted devices accessing the network through the ASR gateways. For this, offline TAC-device mappings are fed to MUR, and MUR generates the signature databases based on EDRs generated by the ASR chassis for various TAC groups.

If MUR is not deployed, then the database file must be manually placed on the ASR chassis under the `/mnt/hd-raid/data/databases/` directory, and loaded into configuration using CLI command.

## Tethering Detection Databases

The Tethering Detection feature uses the OS signature, UA signature, and TAC databases.

These database files must be populated and loaded on to the ASR chassis by the administrator. The procedure to load the databases is the same for all the three types of databases.

Before the database(s) can be loaded for the first time, tethering detection must be enabled using the **tethering-database** CLI command in the Active Charging Service Configuration Mode.

For all three databases, only a full upgrade of a database file is supported. Incremental upgrade is not supported. If, for any particular database, the upgrade procedure fails, the system will revert back to the previous working version of that database.

### OS Signature Database

The OS signature database file is named “os-db”. The file contains OS fingerprint signatures that have been identified as non-smartphone signatures.

The OS fingerprint signature string is a null-terminated ASCII string of maximum 32 bytes in the following format:

`<tl>|<ttl>|<d>|<wlen>|<mss>|<wss>|STEN`

Where:

- *tl*: Total IP Packet Length
- *ttl*: Initial TTL
- *d*: IP DF bit
- *wlen*: TCP Window Length
- *mss*: TCP Maximum Segment Size
- *wss*: TCP option Window Size Scale
- *S*: TCP option Selective ACK OK
- *T*: TCP option Timestamp
- *E*: TCP option EOL
- *N*: TCP option NOP (count)

The maximum number of entries permitted in the os-db file is 16384.

The maximum size of the os-db file can be 524KB + 50 bytes for header and trailer.

In the 12.2 release, the file is in plain text format and contains one TCP signature in ASCII format, one entry per line.

The following is the content of a sample os-db file:

```
Version 1.1

BEGIN OS-DB

48|128|1|5840|1460|1|1112

44|128|0|5840|1460|1|1011

END OS-DB
```



## UA Signature Database

The UA signature database file is named “ua-db”. The file contains UA signatures that have been identified as non-smartphone signatures.

The UA signatures are stored in plain text format in the database file so that manual modification of the database is possible.

The maximum number of entries permitted in the ua-db file is 16384.

The maximum size of the ua-db file can be 67MB + 50 bytes for header and trailer.

The following is the content of a sample ua-db file:

```
Version 1.1
```

```
BEGIN UA-DB
```

```
Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; WOW64; Trident/4.0; SLCC2;  
.NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0;  
InfoPath.2)
```

```
END UA-DB
```

## TAC Database

The TAC database file is named “tac-db”. The file contains smartphone TACs that are uploaded in MUR by the operator.

The maximum number of entries permitted in the tac-db file is 16384.

The maximum size of the tac-db file can be 147KB + 50 bytes for header and trailer.

The following is the content of a sample tac-db file:

```
Version 1.1
```

```
BEGIN TAC-DB
```

```
01194800
```

```
01194801
```

```
END TAC-DB
```

## Loading and Upgrading Tethering Detection Databases

This section provides an overview of loading and upgrading the OS, UA, and TAC databases used in tethering detection.

The database files from MUR must be copied onto the ASR chassis to the following directory path designated for storing the database files:

```
/mnt/hd-raid/databases/
```

Any further upgrades to the database files can be done by placing the file named `new-filename` in the designated directory path. ACS auto-detects the presence of files available for upgrade daily. When a new version of a file is found, the upgrade process is triggered. The upgrade can also be forced by running the upgrade command in the CLI. On a successful upgrade this file is renamed to `filename`.

## Session Recovery Support

In the 12.2 release, the following Session Recovery features are implemented:

- Database recovery after SessCtrl getting killed.
- Database recovery after one or more SessMgrs getting killed.

Note that it may take sometime (ranging from 5 seconds to 5 minutes) for the database to become available in all the SessMgrs post recovery/migration depending on the size of the database files and the number of SessMgrs operational in the system.

## Limitations and Dependencies

This section identifies limitations and dependencies for the Tethering Detection feature.

- The Tethering Detection feature does not cover Network Behind Mobile Subscriber (NBMS) scenarios. That is, it does not distinguish traffic originating from a device behind a smartphone from that originating from the smartphone just by looking at the source IP address and source port. In the absence of NBMS feature, no extra IP addresses are given to smartphone and hence all traffic ingressing into the ASR chassis will have the same source IP address no matter whether it originated from the smartphone or the device connected behind it.
- UA strings exhibited by browser software on smartphones are different than those exhibited by browser software on the laptop/desktop operating systems. Same is the assumption in case of OS signatures. It is assumed that the smartphone OS stacks will emit different characteristics of TCP/IP configuration than that exhibited by desktop OS.
- If a device, such as iPad, has the same OS as that of iPhone, and the OS signatures of the two are identical, then ECS will not be able to detect a tethering session originating from iPad behind the iPhone.
- If a subscriber modifies OS signature as well as UA string of the laptop behind MS in order to pose as a legitimate user, ECS will not be able to detect tethering.

## Tethering Detection Feature Configuration

This section describes how to configure the Tethering Detection feature to detect subscriber flows from PC devices tethered to mobile smartphones.

To enable and configure the Tethering Detection feature, use the following configuration:

**configure**

```
    active-charging service <ecs_service_name>

        tethering-database [ os-signature <os_signature_db_file_name> | tac
<tac_db_file_name> | ua-signature <ua_signature_db_file_name> ] +

        ruledef <tethering_detection_ruledef_name>

            tethering-detection { flow-not-tethered | flow-tethered }

            exit

        rulebase <rulebase_name>

            tethering-detection [ os-db-only | ua-db-only ]

            action priority <priority> ruledef <tethering_detection_ruledef_name>
charging-action <charging_action_name>

            ...

        end
```

## Upgrading Tethering Detection Databases

To upgrade the Tethering Detection feature databases, in the Exec mode, use the following CLI command:

```
upgrade tethering-detection database { all | os-signature | tac | ua-signature }
[ -noconfirm ]
```

## Sample Configurations

The following examples illustrate two different implementations of the Tethering Detection feature's configuration.

- The following type of configuration is suitable where ECS performance is critical and the operator wants to put in a flat charging plan in place for all the tethered traffic. In such a scenario, addition of a single new ruledef to the configuration suffices. Placing this ruledef at the highest priority in the rulebase will ensure all the tethered flows are charged as per the tariff plan for tethered traffic.

```
configure
  active-charging service ecs_service
    tethering-database
    ruledef tethered-traffic
      tethering-detection flow-tethered
      tcp any-match = TRUE
    exit
  ruledef ftp-pkts
    ftp any-match = TRUE
  exit
  ruledef http-pkts
    http any-match = TRUE
  exit
  ruledef tcp-pkts
    tcp any-match = TRUE
  exit
  ruledef ip-pkts
    ip any-match = TRUE
  exit
  ruledef http-port
    tcp either-port = 80
    rule-application routing
  exit
```

```
ruledef ftp-port
    tcp either-port = 21
    rule-application routing
    exit
charging-action premium
    content-id 1
    retransmissions-counted
    billing-action egcdr
    exit
charging-action standard
    content-id 2
    retransmissions-counted
    billing-action egcdr
    exit
rulebase consumer
    tethering-detection
        action priority 10 ruledef tethered-traffic charging-
action premium
        action priority 20 ruledef ftp-pkts charging-action
standard
        action priority 30 ruledef http-pkts charging-action
standard
        action priority 40 ruledef tcp-pkts charging-action
standard
        action priority 50 ruledef ip-pkts charging-action
standard
        route priority 80 ruledef http-port analyzer http
        exit
rulebase default
end
```

- The following type of configuration is suitable when operators want to apply differentiated charging to various flows that are found to be tethered. In this case, traffic that requires different charging action or content ID when it is tethered will be identified using two ruledefs, one with “flow-is-tethered = TRUE” option and another without this option. This configuration provides finer granularity of control but results in higher performance degradation because the rule matching tree size increases.

```
configure

active-charging service ecs_service

tethering-database

ruledef ftp-pkts

    ftp any-match = TRUE

    exit

ruledef ftp-pkts-tethered

    ftp any-match = TRUE

    tethering-detection flow-tethered

    exit

ruledef http-pkts

    http any-match = TRUE

    exit

ruledef http-pkts-tethered

    http any-match = TRUE

    tethering-detection flow-tethered

    exit

ruledef tcp-pkts

    tcp any-match = TRUE

    exit

ruledef tcp-pkts-tethered

    tcp any-match = TRUE

    tethering-detection flow-tethered

    exit

ruledef ip-pkts
```

```
        ip any-match = TRUE
    exit
ruledef ip-pkts-tethered
    ip any-match = TRUE
    tethering-detection flow-tethered
    exit
ruledef http-port
    tcp either-port = 80
    rule-application routing
    exit
ruledef ftp-port
    tcp either-port = 21
    rule-application routing
    exit
charging-action premium-http
    content-id 10
    retransmissions-counted
    billing-action egcdr
    exit
charging-action premium-ftp
    content-id 20
    retransmissions-counted
    billing-action egcdr
    exit
charging-action premium
    content-id 1
    retransmissions-counted
    billing-action egcdr
    exit
```

```
charging-action standard
    content-id 2
    retransmissions-counted
    billing-action egcdr
    exit
rulebase consumer
    tethering-detection
        action priority 10 ruledef ftp-pkts-tethered charging-
action premium-ftp
        action priority 20 ruledef ftp-pkts charging-action
standard
        action priority 30 ruledef http-pkts-tethered charging-
action premium-http
        action priority 40 ruledef http-pkts charging-action
standard
        action priority 50 ruledef tcp-pkts-tethered charging-
action premium
        action priority 60 ruledef tcp-pkts charging-action
standard
        action priority 70 ruledef ip-pkts-tethered charging-
action premium
        action priority 80 ruledef ip-pkts charging-action
standard
        route priority 80 ruledef http-port analyzer http
    exit
rulebase default
    end
```



# CLI Command Reference

This section provides details of new/modified CLI commands required to configure the Tethering Detection feature.

## ACS Configuration Mode Commands

### tethering-database

This command enables the Tethering Detection feature, and loads the databases from the specified files into the service.

#### Product

ACS

#### Privilege

Security Administrator, Administrator

#### Syntax

```
tethering-database [ os-signature os_signature_db_file_name | tac
tac_db_file_name | ua-signature ua_signature_db_file_name ] +
{ default | no } tethering-database
```

---

#### **default**

Configures this command with the default setting.

Default: Tethering Detection feature is disabled, and the database file names are reset to their default values.

---

#### **no**

Disables Tethering Detection.

---

#### **os-signature** *os\_signature\_db\_file\_name*

Specifies name of the OS Signature database file to load.

*os\_signature\_db\_file\_name* must be an alpha and/or numeric string of 1 through 255 characters in length.

---

#### **tac** *tac\_db\_file\_name*

Specifies name of the TAC database file to load.

*tac\_db\_file\_name* must be an alpha and/or numeric string of 1 through 255 characters in length.

---

#### **ua-signature** *ua\_signature\_db\_file\_name*

Specifies name of the User Agent (UA) Signature database file to load.

*ua\_signature\_db\_file\_name* must be an alpha and/or numeric string of 1 through 255 characters in length.

---

#### **+**

Indicates that more than one of the preceding option can be entered in a single command.

---

**Usage**

Use this command to enable the Tethering Detection feature, and load the OS, TAC, and UA databases from the specified files into the service.

Tethering refers to the use of a smartphone as a USB dongle/modem to provide Internet connectivity to laptops/PDAs/tablets like iPad, using the smartphone's data plan. Typically many operators have in place an eat-all-you-can-get data plan for smartphones, the usage of which is intended to be from the smartphone as a mobile device. However, some users use the low rate/unlimited usage of data plan to provide Internet connectivity to their laptops in places where normal Internet connection via broadband/WiFi might be more costly/not available/insecure.

Operators are interested in detecting such usage of a smartphone as a modem to better understand the usage across their networks and offer plans inline to that usage to their customers. They may also charge the tethered and non-tethered traffic separately.

After Tethering Detection has been enabled here (regardless, it must also be enabled within the rulebase), this CLI command may be used to change the databases with the specified databases.

The files are picked from the disk file system within the /databases folder. If a file name value is not configured, the default file names, *os-db*, *tac-db*, and *ua-db*, are used.

---

**Example**

The following command enables Tethering Detection and selects the UA Signature database file named *test*:

```
tethering-database ua-signature test
```

## ACS Rulebase Configuration Mode Commands

### tethering-detection

This command enables/disables the Tethering Detection feature for a rulebase, and configures the database to use.

**Product**

ACS

**Privilege**

Security Administrator, Administrator

**Syntax**

```
tethering-detection [ os-db-only | ua-db-only ]  
{ default | no } tethering-detection
```

---

**default**

Configures this command with the default setting.

Default: By default, the Tethering Detection feature is disabled. When enabled, by default Tethering Detection will make use of both the databases, unless a specific database is specified to be used.

---

**no**

Disables the Tethering Detection feature for the rulebase.

---

**os-db-only**

Specifies to perform tethering detection using only the OS database.

---

**ua-db-only**

Specifies to perform tethering detection using only the UA database.

---

**Usage**

Use this command to enable/disable the Tethering Detection feature for a rulebase, and configure the database to use.

Changing the configuration does not affect existing flows of the subscriber. If Tethering Detection was disabled and is turned enabled, it will be applied only to new flows of subscribers using the rulebase. Also, see the **tethering-database** command in the *ACS Configuration Mode Commands* chapter.

---

**Example**

The following enables the Tethering Detection feature in the rulebase, and specifies to use only the OS database:

```
tethering-detection os-db-only
```

## ACS Ruledef Configuration Mode Commands

### tethering-detection

This command defines rule expressions to match tethered/non-tethered flows.

**Product**

All

**Privilege**

Security Administrator, Administrator

**Syntax**

```
tethering-detection { flow-not-tethered | flow-tethered }
```

```
no tethering-detection
```

---

**no**

Deletes the tethering detection configuration from the ruledef.

---

**flow-not-tethered**

Specifies to match if tethering is not detected on flow.

---

**flow-tethered**

Specifies to match if tethering is detected on flow.

---

**Usage**

Use this command to define rule expressions to match tethered/non-tethered flows.

Note that in order for the rule containing the tethering-detection configuration to get matched, at least one valid rule line has to be present in it.

This configuration is treated in a special manner by the rule matching engine in a way that it is excluded from the condition **multi-line-or all-lines**. For example, if there are three rule-lines in a ruledef and multi-line-or is enabled as follows:

```
ruledef all-tethered-web-traffic

  http any-match = TRUE

  wsp any-match = TRUE

  multi-line-or all-lines

  tethering-detection flow-tethered

  exit
```

In this case, if for a packet only the rule line **tethering-detection flow-tethered** matches, it is not sufficient to result in a rule match even though **multi-line-or all-lines** is enabled in the ruledef.

---

**Example**

The following command defines a rule expression to match tethered flows:

```
tethering-detection flow-tethered
```

## EDR Format Configuration Mode Commands

### rule-variable

This command specifies the order of fields in the EDR.

#### Product

All

#### Privilege

Security Administrator, Administrator

#### Syntax

```
rule-variable protocol rule priority priority [ in-quotes ]
```

```
no rule-variable protocol rule [ priority priority ]
```

---

**no**

Removes the previously configured rule variable protocol rule and/or priority for EDR attribute.

---

*protocol rule*

Specifies the rule variable for EDR format.

*protocol* must be one of the following with specified rule:

- **flow**: Flow related fields:
  - **tethered**: Tethering detected on flow. Enables/disables tethering detection result field in EDRs sent to MUR.
- **tcp**: Transmission Control Protocol (TCP) related fields:
  - **os-signature**: OS signature string for TCP flow. Enables/disables OS Signature field in EDRs sent to MUR.



**Important:** For more information on other options available with this command, see the *EDR Format Configuration Mode Commands* chapter in the *Command Line Interface Reference*.

---

**priority** *priority*

Specifies the CSV position of protocol rule related information in EDR.

*priority* must be an integer from 1 through 65535.

---

#### Usage

Use this command to set what field appears where—in which order—in the EDR.

A particular field in an EDR format can be entered multiple times at different priorities. While removing the EDR field using the **no rule-variable** command you can remove all occurrences of a particular field by specifying the field name or a single occurrence by additionally specifying the optional **priority** keyword.

---

**Example**

The following is examples of the command:

```
rule-variable tcp os-signature priority 36
```

```
rule-variable flow tethered priority 37
```

## Exec Mode Commands

### clear active-charging tethering-detection statistics

This command clears statistics pertaining to the Tethering Detection feature.

**Product**

ACS

**Privilege**

Security Administrator, Administrator, Operator, Inspector

**Syntax**

```
clear active-charging tethering-detection statistics
```

---

**Usage**

Use this command to clear statistics pertaining to the Tethering Detection feature.



## show active-charging tethering-detection

This command displays information/statistics pertaining to Tethering Detection databases.

### Product

ACS

### Privilege

Security Administrator, Administrator, Operator, Inspector

### Syntax

```
show active-charging tethering-detection { database [ os-signature | tac | ua-  
signature ]+ [ sessmgr { all | instance instance } ] [ | { grep grep_options |  
more } ] | statistics }
```

---

```
database [ os-signature | tac | ua-signature ]+ [ sessmgr { all |  
instance instance } ]
```

Displays information pertaining to the specified Tethering Detection database(s).

- **os-signature**: Displays Tethering Detection OS database information.
- **tac**: Displays Tethering Detection TAC database information.
- **ua-signature**: Displays Tethering Detection UA database information.
- **+**: Indicates that more than one of the preceding keywords can be entered in a single command.
- **sessmgr { all | instance *instance* }**: Displays SessMgr Tethering Detection database status.
  - **all**: Displays status for all SessMgr instances.
  - **instance *instance***: Displays status for the specified SessMgr instance.  
*instance* must be an integer from 1 through 10000.

---

**statistics**

Displays Tethering Detection related statistics.

---

```
grep grep_options | more
```

Specifies that output of this command is to be piped (sent) to the command specified. A command to send the output to must be specified.

---

### Usage

Use this command to view information/statistics pertaining to Tethering Detection databases.

---

**Example**

The following command displays information pertaining to Tethering Detection UA and OS databases:

```
show active-charging tethering-detection database ua-signature os-  
signature
```

The following command displays information pertaining to all Tethering Detection databases:

```
show active-charging tethering-detection database
```

## upgrade tethering-detection

This command upgrades the Tethering Detection feature's database(s).

### Product

ACS

### Privilege

Security Administrator, Administrator

### Syntax

```
upgrade tethering-detection database { all os-signature | tac | ua-signature } [
-noconfirm ]
```

---

#### **all**

Specifies to upgrade all Tethering Detection databases—OS, TAC, and UA.

---

#### **os-signature**

Specifies to upgrade the OS database.

---

#### **tac**

Specifies to upgrade the TAC database.

---

#### **ua-signature**

Specifies to upgrade the UA database.

---

#### **- noconfirm**

Specifies that the command must execute without any prompts and confirmation from the user.

---

### Usage

Use this command to upgrade the database(s) used by the Tethering Detection feature.

This command upgrades the database(s) from file(s) kept in designated path. The name of the existing source file is prefixed with the word “new-”. For example for OS DB, if the existing file name is “os-db”, the upgrade file name is “new-os-db”.

If there is a file named “new-xxx-db”, it is verified that it is a valid Tethering Detection database and then loaded it into memory. If successful, the file is renamed “xxx-db” to “xxx-db-<number>” and then renamed “new-xxx-db” to “new-xxx-db”.

For example, the command **upgrade tethering-database ua-signature -noconfirm** results in loading the file by name “new-ua-db” if it is present in the designated directory. In case of a successful upgrade, the previous version of the database is stored as backup in a file named “ua-db-1”. Also, the newly uploaded database file is renamed as “ua-db”.

---

### Example

The following command upgrades all Tethering Detection databases:

```
upgrade tethering-detection database all -noconfirm
```

