DNS Snooping

This chapter describes the DNS Snooping feature and provides detailed information on the following topics:

Feature Summary and Revision History

Summary Data

Applicable Product(s) or Functional Area	ECS
Applicable Platform(s)	• ASR 5500 • VPC-DI • VPC-SI
Feature Default	Disabled - License Required
Related Changes in This Release	Not applicable
Related Documentation	*Command Line Interface Reference*

Revision History

Important Revision history details are not provided for features introduced before releases 21.2 and N5.1.

Revision Details	Release
First introduced.	Pre 21.2

Feature Description

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 is 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 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.

License Requirements

DNS Snooping is a licensed Cisco feature. A separate feature license may be required. Contact your Cisco account representative for detailed information on specific licensing requirements. For information on installing and verifying licenses, refer to the *Managing License Keys* section of the *Software Management Operations* chapter in the *System Administration Guide*.

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. Until 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 must define 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. Also, in case subscribers make DNS queries to their self-created DNS servers and hack the response being sent, it can result in inclusion of invalid IP addresses in the list. In this case, the IP addresses will be learnt and the traffic may be free-rated or blocked incorrectly depending on the action set. Therefore the above is suggested to avoid attacks on DNS traffic.
- There is a limit on the 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 is a limit on the total number of server-domain-name ruledefs that can be configured.
- If same IP address is returned in DNS responses for different DNS q-names (same IP hosting multiple URLs), than while rule matching, the higher priority rule having this learnt-IP address will be matched. This can have undesired rule-matching as explained next.

  For example, if DNS queries for both www.facebook.com and www.cnn.com returned the 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.
- In the 12.2 release, the lookup for IPv6 learnt IP addresses will not be optimized. Hash based lookup (optimization) is done for IPv4 address lookup. In a later release Longest Prefixed Match (LPM) based optimization will be considered for both IPv4 and IPv6 learnt IP address matching.

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 CNAME. 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 following are the maximum limits:

- 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.

In 12.3 and earlier releases, the CLI command **show active-charging dns-learnt-ip-addresses statistics sessmgr all** displayed all the configured patterns and rulebase names for each pattern entry, even though the pattern has not learnt any IP address.

When a large number of DNS snooping ruledefs are configured (configured as ip server-domain name under ruledef configuration), the memory allocated for sending this information exceeds the message size limit for messenger calls and hence the crash is observed.

In 14.0 and later releases, the **show active-charging dns-learnt-ip-addresses statistics sessmgr all** CLI command will be displaying only the patterns for which at least one IPv4/IPv6 address is learnt as all other information is available from the configuration.

The following call flow illustration and descriptions explain the working of the DNS Snooping feature.

Figure 1: DNS Snooping Call Flow

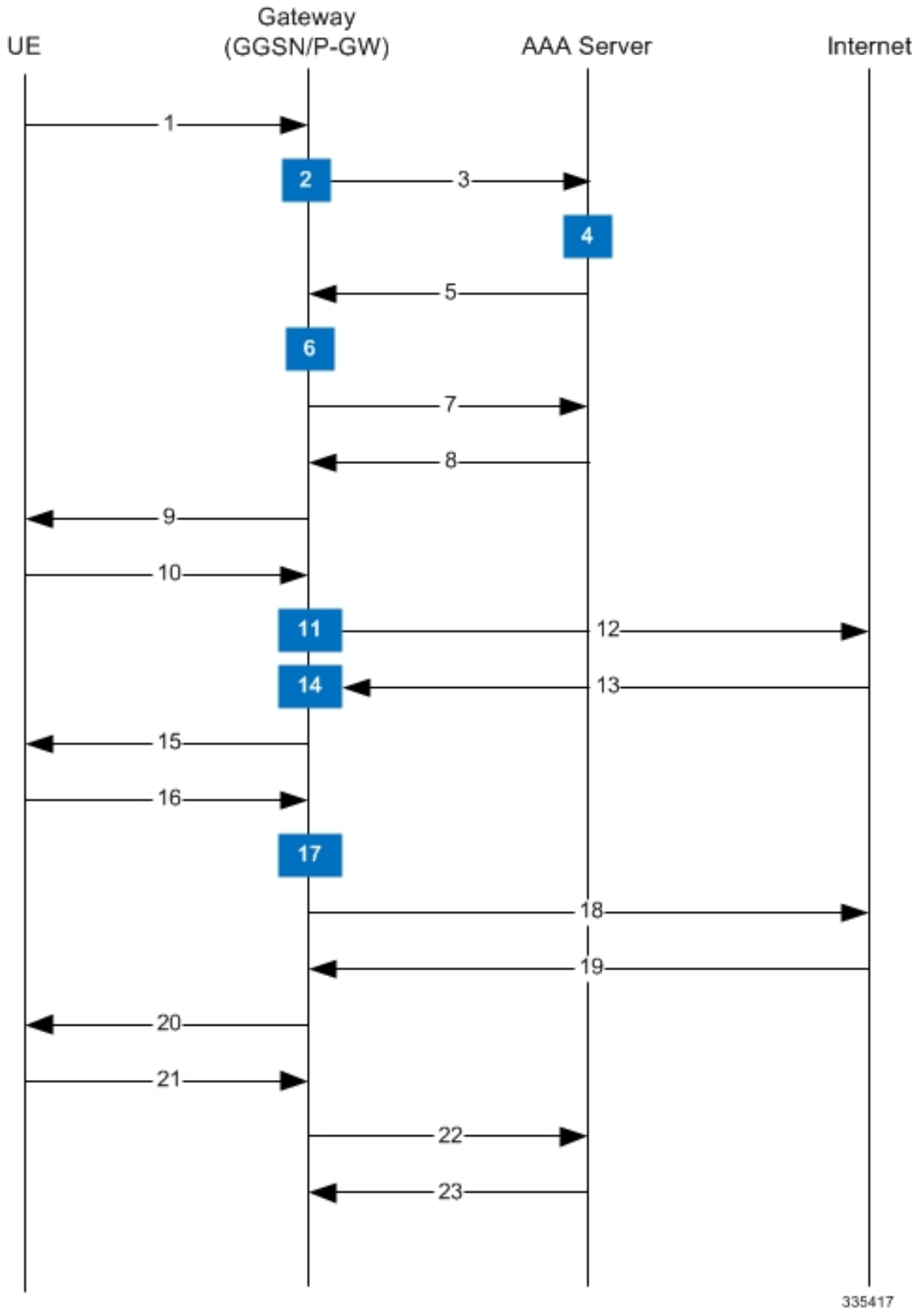
Gateway
(GGSN/P-GW)
UE
AAA Server
Internet
1
2
3
4
5
6
7
8
9
10
11
12
14
13
15
16
17
18
19
20
21
22
23
335417

Table 1: DNS Snooping Call Flow Descriptions

Step No.	Description
1	UE requests the system for registration.
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 (usernamedomain), Calling Station ID (IMSI, MSID), and Framed IP Address (HoA) as the basis for subscriber lookup.
5	The Policy Manager and AAA generate and send an Access Accept message including all policy and other attributes to establish the session to ECS. The Policy Manager and/or AAA include following attributes in the Access Accept message: • 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, 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. • 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 configuration is defined in the ACS Configuration Mode and can be applied to individual subscribers, domains, or on per-context basis.
6	ECS creates a new session for UE, and sends the rulebase to ACS subsystem if required.

Step No.	Description
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 analyzes the query-name from the subscriber's DNS query, 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.
17	Looking at the server-ip-address of the 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.

Configuring DNS Snooping

Use the following configuration to configure the DNS Snooping feature:

```
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
```

Verifying the DNS Snooping Configuration

Enter the following command to check the number of DNS learnt IP-entries per ruleline.

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

Monitoring and Troubleshooting the DNS Snooping feature

This section provides information regarding bulk statistics, show commands and/or their outputs in support of this feature.

show active-charging dns-learnt-ip-addresses statistics sessmgr instance <instance> verbose

The following fields display the statistics related to the DNS Snooping feature.

- Sessmgr Instance
- Pattern
- Rulebase
- List of CNAMES
- Destination Context
- Total-ipv4-entries
- Ipv4-Entries-flushed
- Ipv4-TTL-replaced

- Ipv4-Overflows
- Total-ipv6-entries
- Ipv6-Entries-flushed
- Ipv6-TTL-replaced
- Ipv6-Overflows
- Ipv4 Address TTL (in secs)
- Ipv6 Address TTL (in secs)
- Summary:
  - Total learnt ipv4 entries
  - Total learnt ipv6 entries

Bulk Statistics

Bulk statistics reporting for the DNS Snooping feature is supported.

The following bulk statistics are available in the ECS schema:

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