

X-Header Insertion and Encryption

- Feature Summary and Revision History, on page 1
- Feature Description, on page 2
- Supported Encryption Methods, on page 7
- How It Works, on page 8
- Configuring X-Header Insertion and Encryption, on page 9
- Monitoring and Troubleshooting the X-Header Insertion and Encryption feature, on page 12

Feature Summary and Revision History

Table 1: Summary Data

Applicable Products and Functional Area	• P-GW • SAEGW
Applicable Platforms	• ASR 5500 • VPC - DI • VPC - SI
Feature Default	Disabled - License Required
Related Changes in This Release	Not applicable
Related Documentation	• *Command Line Interface Reference* • *ECS Administration Guide* • *Statistics and Counters Reference*

Revision History

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

Revision Details	Details
In this release, the TLS_RSA_WITH_AES_256_GCM_SHA_384 algorithm is introduced for enhanced security.	21.9
First introduced.	Pre 21.2

Feature Description

The X-Header Insertion and X-Header Encryption features, collectively known as Header Enrichment, enables to append headers to HTTP/WSP GET and POST request packets, and HTTP Response packets for use by end applications, such as mobile advertisement insertion (MSISDN, IMSI, IP address, user-customizable, and so on).



Important In this release, the X-Header Insertion and Encryption features are supported only on the GGSN and P-GW.

Following are the software requirements for the new TLS_RSA_WITH_AES_256_GCM_SHA_384 attribute in CDR:

- Configure AES-256-GCM-sha384 encryption algorithm with 256-bit keys. This configuration is same as the one used for the RC4MD5 encryption.
- Use the existing re-encryption timeout CLI, which is used at rulebase level, for re-encryption.
- For AES-GCM encryption, use the optional **salt** flag. This flag is used to randomize the keys, which are generated from the passphrase, and the Initialization Vectors (IV).

License Requirements

X-Header Insertion and X-Header Encryption are both licensed Cisco features. 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*.

X-Header Insertion

This section provides an overview of the X-Header Insertion feature.

Extension header (x-header) fields are the fields not defined in RFCs or standards but can be added to headers of protocol for specific purposes. The x-header mechanism allows additional entity-header fields to be defined without changing the protocol, but these fields cannot be assumed to be recognizable by the recipient. Unrecognized header fields should be ignored by the recipient and must be forwarded by transparent proxies.

The X-Header Insertion feature enables inserting x-headers in HTTP/WSP GET and POST request packets and HTTP response packets. Operators wanting to insert x-headers in HTTP/WSP request and HTTP response packets, can configure rules for it. The charging-action associated with the rules will contain the list of x-headers to be inserted in the packets.

For example, if you want to insert the field *x-rat-type* in the HTTP header with a value of *rat-type*, the header inserted should be:

x-rat-type: geran

where, *rat-type* is *geran* for the current packet.

X-Header Encryption

This section provides an overview of the X-Header Encryption feature.

X-Header Encryption enhances the X-header Insertion feature to increase the number of fields that can be inserted, and also enables encrypting the fields before inserting them.

If x-header insertion has already happened for an IP flow (because of any x-header format), and if the current charging-action has the first-request-only flag set, x-header insertion will not happen for that format. If the first-request-only flag is not set in a charging-action, then for that x-header format, insertion will continue happening in any further suitable packets in that IP flow.

Changes to x-header format configuration will not trigger re-encryption for existing calls. The changed configuration will however, be applicable for new calls. The changed configuration will also apply at the next re-encryption time to those existing calls for which re-encryption timeout is specified. If encryption is enabled for a parameter while data is flowing, since its encrypted value will not be available, insertion of that parameter will stop.



Important Recovery of flows is not supported for this feature.

TCP OOO Packets

ECS handles TCP OOO packets in two ways depending on the rulebase configuration:

Transmit Immediately: If the rulebase is configured to transmit immediately for TCP OOO packets, the OOO packets will be forwarded immediately, and a copy of this packet will be added to the OOO queue for analysis.

Transmit After Reordering: If the rulebase is configured to transmit after reordering for TCP OOO packets, the OOO packets will be added to the OOO queue for analysis. Header insertion on OOO request packets occurs on reordering packets that are received before the OOO request timeout. When a reordering packet is received, the queued packets are forwarded. However, if a reordering packet is not received before the OOO queue timeout, the queued packet will be forwarded without any analysis done to those packets.



Important When TCP OOO processing has been configured in the rulebase, a session manager crash might be observed due to overlapping TCP segments and/or reordering packet arriving within TCP OOO configured timeout value or default value (5 sec). This issue can be resolved by changing the rulebase configuration for TCP OOO packets from **transmit after-reordering** to **transmit immediately**.

In 20 and later releases, TCP OOO packets will be buffered for HTTP traffic until the header enrichment is completed. The header enrichment is supported on either first request packet or all request packets, response packets, or on both. So following packets after the header enrichment is complete will not require buffering of OOO packets and the packets can be transmitted immediately. This will improve memory optimization

and network performance. The out-of-order packets will be buffered as per the x-header configuration in any of the charging-action for the subscriber's rulebase.

- If x-header insertion is only for the request packet, then out-of-order buffering will be supported till the header completion of the request packet.
- If x-header insertion is only for the response packet, then out-of-order buffering will be supported till the header completion of the response packet.
- If x-header insertion is for both request and response packets, then out-of-order buffering will be supported till the completion of HTTP headers for that packet.

Limitations of buffering TCP OOO packets:

- This enhancement will be supported only for HTTP flows of x-header enrichment feature.
- In case of pipeline flows with multiple transactions, if a new OOO request/response is received while the previous request/response is still going on, then x-header insertion will not work for the new request/response of that flow.

IP Fragmented Packets

ECS can perform Header Enrichment to IP fragmented packets when all the fragments are received before the reassembly timeout. If the packet size after Header Enrichment exceeds the MSS of the session, the reassembled packet gets segmented, the multiple segments are forwarded.

Limitations to the Header Insertion and Encryption Features

This section lists known limitations to insertion and encryption of x-header fields in HTTP/WSP request and HTTP response packets.

The following are limitations to insertion and encryption of x-header fields in HTTP headers.

Limitations in StarOS 14.0 and later releases:

- Header insertion does not occur for packets with incomplete HTTP headers.
- If a flow has x-header insertion and later some IP fragments are received for which reassembly fails, sequence space of that segment will be mismatched.
- ECS does not support applying more than one modifying action on an inbound packet before sending it on the outbound interface. For example, if header insertion is applied on a packet, then the same packet is not allowed to be modified for NAT/ALG and MSS insertion.
- Header enrichment works only for the first request of a packet with concatenated requests, when the packets are buffered at DCCA. There are no limitations on header enrichment for single GET or pipelined GET requests.
- Header enrichment works for packets at DCCA only when the packets pending of header insertion is buffered.
- Receive window will not be considered during header enrichment. That is, after header enrichment if packet exceeds receive window, ECS will not truncate the packet.
- The maximum bytes per request after header enrichment is 2400 bytes. If concatenated requests exist, a maximum of 2400 bytes after header enrichment can be inserted.

If due to header insertion, the packet size exceeds this limit, the behavior is unpredictable.

- Only those x-header fields in header portion of application protocol that begin with "x-" are parsed at HTTP analyzer. In URL and data portion of HTTP any field can be parsed.
- EDR generation for x-header fields in Response packets will not be supported.

Limitations in StarOS 12.3 and earlier releases:

- The packet size is assumed to be less than "Internal MED MTU size, the size of header fields inserted". If the total length of packet exceeds the internal MTU size, header insertion will not occur after the addition of fields.
- Header insertion occurs for both HTTP GET and POST requests. However, for POST requests, the resulting packet size will likely be larger than for GET requests due to the message body contained in the request. If the previous limitation applies, then POST request will suffer a bigger limit due to this.
- Header insertion does not occur for retransmitted packets.
- Header insertion does not occur for packets with incomplete HTTP headers.
- Header insertion does not occur for TCP OOO and IP fragmented packets.
- If a flow has x-header insertion and later some IP fragments are received for which reassembly fails, sequence space of that segment will be mismatched.
- ECS does not support applying more than one modifying action on an inbound packet before sending it on the outbound interface. For example, if header insertion is applied on a packet, then the same packet is not allowed to be modified for NAT/ALG and MSS insertion.
- If a packet is buffered by ICAP, header insertion will not occur for that packet.
- Receive window will not be considered during header enrichment. That is, after header enrichment if packet exceeds receive window, ECS will not truncate the packet.
- Packet size limit is 2400 bytes, if due to header insertion packet size exceeds this limit, behavior is unpredictable.
- Only those x-header fields in header portion of application protocol that begin with "x-" are parsed at HTTP analyzer. In URL and data portion of HTTP any field can be parsed.

The following are limitations to insertion and encryption of x-header fields in WSP headers:

- x-header fields are not inserted in IP fragmented packets.
- In case of concatenated request, x-header fields are only inserted in first GET or POST request (if rule matches for the same). X-header fields are not inserted in the second or later GET/POST requests in the concatenated requests. For example, if there is ACK+GET in packet, x-header is inserted in the GET packet. However, if GET1+GET2 is present in the packet and rule matches for GET2 and not GET1 x-header is still inserted in GET2. In case of GET+POST also, x-header is not inserted in POST.
- In case of CO, x-header fields are not inserted if the WTP packets are received out of order (even after proper re-ordering).
- If route to MMS is present, x-headers are not inserted.

- x-headers are not inserted in WSP POST packet when header is segmented. This is because POST contains header length field which needs to be modified after addition of x-headers. In segmented WSP headers, header length field may be present in one packet and header may complete in another packet.
- x-headers are not inserted in case of packets buffered at DCCA.

Supported X-Headers

This section provides information on the different x-headers supported by ECS.

ECS supports insertion of various x-header fields in the HTTP/WSP GET and POST request packets and HTTP response packets. The x-headers are inserted at the end of the HTTP/WSP header.

The following bearer-related x-headers are supported:

- 3gpp

  The following 3GPP associated fields are supported:

  - apn
  - charging-characteristics
  - charging-id
  - imei
  - imsi
  - qos
  - rat-type
  - s-mcc-mnc
  - sgsn-address

- acr
- customer-id
- ggsn-address
- mdn
- msisdn-no-cc
- radius-string
- radius-calling-station-id
- session-id
- sn-rulebase
- subscriber-ip-address
- username
- user-profile
- uli

The following HTTP-related x-headers are supported:

- host
- url

In addition, ECS also allows string constants to be inserted as an x-header. For more information on configuring the x-header formats, see the *insert* command section in the *ACS x-Header Format Configuration Mode Commands* chapter of the *Command Line Interface Reference*.

X-Header Enrichment Anti-Spoofing

This section provides an overview of the x-Header Enrichment Anti-Spoofing feature.

The Header Enrichment feature allows operators to encrypt and insert subscriber-specific fields as x-headers in to the HTTP headers of URL requests. However, this might leave the header open to spoofing by malicious external devices. The Anti-Spoofing feature enables deletion and modification of the existing x-header fields to protect the operators and subscribers from spoofing, and provides a fraud detection mechanism when an external portal is used for a subscriber or content authorization.

The feature detects and removes user-generated HTTP headers if the header name is similar to the header name used in the x-header format, and when multiple entries of the same field exist in the header, all the similar entries are removed and one with a modified value is inserted at the end of the HTTP header.

When anti-spoofing is enabled, and if the HTTP header in the GET or POST request spawns across more than one packet, the packets with incomplete HTTP header will be buffered. The buffered packets are sent out once the HTTP header is completed.

The Anti-Spoofing feature is disabled by default and can be enabled/disabled at a field level in the CLI.

Limitations to the Anti-Spoofing Feature

- Header enrichment does not occur if a route to the MMS analyzer exist in the rulebase.
- Header enrichment works only for the first request of a packet with concatenated requests, when the packets are buffered at DCCA.
- If a packet is buffered by ICAP, header insertion will not occur for that packet.
- ECS will not be able to perform header enrichment when all fragments are not received before reassembly timeout in the case of IP fragments packets.
- ECS does not perform more than one flow action which modifies the inbound packet before sending it on the outbound interface.
- If the HTTP GET or POST header is not completed in three packets, anti-spoofing will occur only for the last packet in which the header completes, as buffering supported only up to a maximum of two packets.
- Though insertion of fields is allowed without having "x-" in the field name, extension header fields that do not start with "x-" are not deleted.
- The anti-spoofing feature will not be supported for x-headers inserted in Response messages.

Supported Encryption Methods

In Release 21.9, the TLS_RSA_WITH_AES_256_GCM_SHA_384 algorithm is introduced for enhanced security.

The supported types of encryption for encrypting x-header values are RSA, RC4MD5, and AES-256-GCM-SHA384. These encryption types are explained in the following sections.

RSA

You can configure RSA encryption by using an encryption certificate. With this encryption, during a call, the configured fields of the **xheader-format** command are encrypted. When the charging action is hit for the traffic, then the encrypted values of the configured fields are inserted in the HTTP header.

Important The encrypted values change only if the re-encryption timer times out. As RSA encryption consumes time, it is not used each time a field value changes during a call.

RC4MD5

You can configure RC4MD5 encryption at charging action level. For traffic, an encryption that is configured at a charging level takes precedence over rulebase.

In RC4MD5 encryption, the MD5 hash of the key, which is a 128-bit value, is used to encrypt the value using the RC4 encryption. The base 64 value of the value that is received from the RC4 encryption is then inserted in the x-header.

When the charging action is hit for the traffic, then the encrypted values of the configured fields are inserted in the HTTP header.

AES-256-GCM-SHA384

You can configure AES-256-GCM-SHA384 encryption at charging action level. For traffic, an encryption that is configured at a charging action level takes precedence over rulebase.

In AES-256-GCM-SHA384 encryption, the SHA384 hash of the key, which is 384 bits value, is used to encrypt the value using the AES-GCM algorithm. The base 64 of this encrypted value is then inserted in the x-header.

How It Works

This section describes the steps involved to configure the X-Header Insertion and X-Header Encryption features.

X-Header Insertion

The following steps describe how X-Header Insertion works:

Step 1 Creating/configuring a ruledef to identify the HTTP/WSP packets in which the x-headers must be inserted.

Step 2 Creating/configuring a rulebase and configuring the charging-action, which will insert the x-header fields into the HTTP/WSP packets.

Step 3 Creating/configuring the x-header format.

Step 4 Configuring insertion of the x-header fields based on the message type in the charging action.

X-Header Encryption

The following steps describe how X-Header Encryption works:

Step 1 X-header insertion, encryption, and the encryption certificate is configured in the CLI.

Step 2 When the call gets connected, and after each regeneration time, the encryption certificate is used to encrypt the strings.

Step 3 When a packet hits a ruledef that has x-header format configured in its charging-action, x-header insertion into that packet is done using the given x-header-format.

Step 4 If x-header-insertion is to be done for fields which are marked as encrypt, the previously encrypted value is populated for that field accordingly.

Configuring X-Header Insertion and Encryption

This section describes how to configure the X-Header Insertion and Encryption features, collectively known as Header Enrichment.

Configuring X-Header Insertion

This section describes how to configure the X-Header Insertion feature.



Important This feature is license dependent. Please contact your Cisco account representative for more information.

To configure the X-Header Insertion feature:

Step 1 Create/configure a ruledef to identify the HTTP packets in which the x-headers must be inserted. For information on how to create/configure ruledefs, see the *Configuring Rule Definitions* section in the *Enhanced Charging Service Configuration* chapter.

Step 2 Create/configure a rulebase and configure the charging-action, which will insert the x-header fields into the HTTP packets. For information on how to create/configure rulebases, see the *Configuring Rulebase* section in the *Enhanced Charging Service Configuration* chapter.

Step 3 Create the x-header format as described in Creating the X-Header Format, on page 9.

Step 4 Configure the x-header format as described in Configuring the X-Header Format, on page 10.

Step 5 Configure insertion of the x-header fields as described in Configuring Charging Action for Insertion of X-Header Fields, on page 10.

Creating the X-Header Format

To create an x-header format, use the following configuration:

```
configure
  active-charging service ecs_service_name
```

```
xheader-format xheader_format_name
end
```

Configuring the X-Header Format

To configure an x-header format, use the following configuration:

```
configure
  active-charging service ecs_service_name
    xheader-format xheader_format_name
      insert xheader_field_name { string-constant xheader_field_value | variable
 { bearer { 3gpp { apn | charging-characteristics | charging-id | imei |
 imsi | qos | rat-type | s-mcc-mnc | sgsn-address } | acr | customer-id
| ggsn-address | mdn | msisdn-no-cc |  radius-string |
radius-calling-station-id | session-id | sn-rulebase |
subscriber-ip-address | username } [ encrypt ] | http { host | url } }
       end
```

Configuring Charging Action for Insertion of X-Header Fields

To configure a charging action for insertion of x-header fields, use the following configuration:

```
configure
  active-charging service ecs_service_name
    charging-action charging_action_name
     xheader-insert xheader-format xheader_format_name [ encryption { rc4md5
 | aes-256-gcm-sha384 [ salt ] } [ encrypted ] key key ] [
first-request-only ] [ msg-type { response-only | request-and-response }
 ] [ -noconfirm ]
       end
```

Note

- If rc4md5 encryption is configured in the charging action, it will take precedence over RSA certificate based encryption for flows hitting particular charging action.
- X-header insertion in HTTP Response packets can be enabled/disabled using the **msg-type** keyword.
  - **response-only**: When configured in charging-action, x-header will be inserted in HTTP Response packets with specified x-header format.
  - **request-and-response**: When configured in charging-action, x-header will be inserted in both HTTP Request and Response packets with same x-header format.

Configuring X-Header Encryption

This section describes how to configure the X-Header Encryption feature.

Important This feature is license dependent. Please contact your Cisco account representative for more information.

To configure the X-Header Encryption feature:

Step 1 Configure X-Header Insertion as described in Configuring X-Header Insertion, on page 9.

Step 2 Create/configure a rulebase and configure the encryption certificate to use and the re-encryption parameter as described in Configuring X-Header Encryption, on page 11.

Step 3 Configure the encryption certificate to use as described in Configuring Encryption Certificate, on page 11.

Configuring X-Header Encryption

To configure X-Header Encryption, use the following configuration example:

```
configure
  active-charging service ecs_service_name
    rulebase rulebase_name
      xheader-encryption certificate-name certificate_name
      xheader-encryption re-encryption period re-encryption_period
      end
```

Notes:

- This configuration enables X-Header Encryption for all subscribers using the specified rulebase *rulebase_name*.
- If the certificate is removed, ECS will continue using the copy that it has. It will only free its copy if the certificate name is removed from the rulebase.
- Changes to x-header format configuration will not trigger re-encryption for existing calls. The changed configuration will however, be applicable for new calls. The changed configuration will also apply at the next re-encryption time to those existing calls for which re-encryption timeout is specified. If encryption is enabled for a parameter while data is flowing, since its encrypted value will not be available, insertion of that parameter will stop.

Configuring Encryption Certificate

To configure the encryption certificate, use the following configuration example:

```
configure
  certificate name certificate_name pem { { data pem_certificate_data private-key
 pem [ encrypted ] data pem_pvt_key } | { url url private-key pem { [
encrypted ] data pem_pvt_key | url url } }
  end
```

Verifying the X-Header Insertion and Encryption Configuration

Enter the following command in the Exec Mode to verify your configuration:

```
show active-charging xheader-format name xheader_format_name
```

Monitoring and Troubleshooting the X-Header Insertion and Encryption feature

This section provides information on the show commands and/or their outputs available to support this feature.

show active-charging charging-action name

The output of this command displays the information for the RSA header enrichment encryption algorithm.

- Encryption Type: aes-256-gcm-sha384
- Salt : YES/NO

show active-charging charging-action statistics name

The output of this command displays statistics for X-header information.

- XHeader Information:
  - XHeader Bytes Injected
  - XHeader Pkts Injected
  - IP Frags consumed by XHeader
  - XHeader Bytes Removed
  - XHeader Pkts Removed

show active-charging rulebase statistics name

The output of this command displays the Header Enrichment statistics.

- HTTP header buffering limit reached