HotSpot Feature

Teldat Dm820-I

Legal Notice

Warranty

This publication is subject to change.

Teldat offers no warranty whatsoever for information contained in this manual.

Teldat is not liable for any direct, indirect, collateral, consequential or any other damage connected to the delivery, supply or use of this manual.

Table of Contents

I Related Documents

Teldat Dm800-I AAA Feature

Teldat Dm733-I Radius Protocol

Teldat Dm752-I Access Control

Teldat Dm739-I IPSec

Teldat Dm723-I DNS

Teldat Dm715-I BRS

Chapter 1 HotSpot feature

1.1 Introduction

In a network, a HotSpot gateway is an element that provides Internet access and other services, typically over wireless technologies, to occasional client devices during a specific period of time.

Intelligent client management may be required when offering additional services, such as establishing user-based session privileges, client statistics accounting, or service agreement requests.

Typical services deployed by a HotSpot gateway require user information, which cannot be received through the communications link or network layers. Thus, specific mechanisms are implemented to gather this essential information. Captive portals are generally used to obtain said information (credentials) from network clients.

A captive portal is a web browser-based tool used to force a client to present user credentials before network access is granted. Since this tool operates at the application layer, the HotSpot has to facilitate appropriate methods for a successful connection at lower layers.

As soon as the HotSpot gateway obtains user credentials, Authentication, Authorization and Accounting (AAA) services can be used to manage clients. As a result, session context maintenance and the implementation of required protocols are often needed to interact with AAA servers.

The HotSpot feature aims to provide a set of functions that help integrate tasks reviewed by a HotSpot gateway.

This document explains how to implement a HotSpot gateway solution using the HotSpot feature. Functions provided by the HotSpot feature are detailed, together with their configuration and monitoring aspects.

1.2 Proprietary solution

Our solution provides a HotSpot gateway that is ready to interact with a captive portal and AAA servers running in external locations. This means the captive portal can be located wherever commercial solutions for captive portals can be tested (even online).

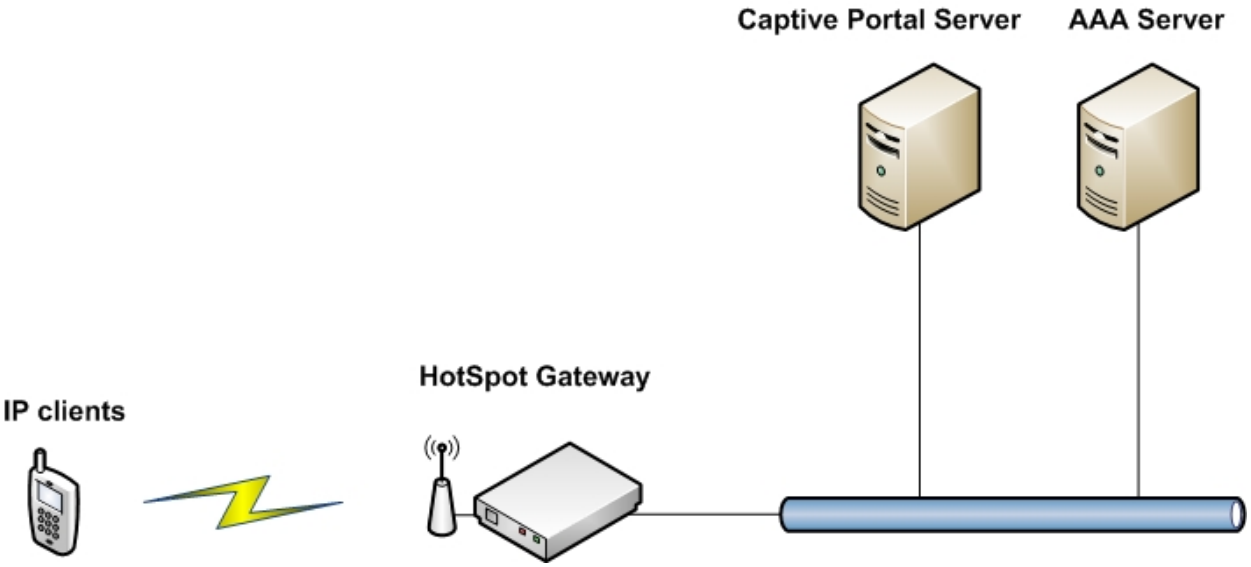

Fig. 1: **Network diagram: HotSpot gateway solution with captive portal authentication**

Main characteristics and services:

- Redirecting HTTP traffic to an external captive portal.
- Interoperability with AAA protocols.
- Client session management: configurable network access policies, uplink/downlink bandwidth limits, maximum credit for uplink/downlink bytes, monitoring of client statistics and periodic session accounting.
- Definition of a *White List* containing MAC addresses that grant network access to clients without requesting user credentials.
- Definition of a *Walled Garden* to control client access to web content and services.

Solution restrictions:

- Every captive portal uses a different format to specify user credentials over HTTP sessions. This lack of homogeneity implies the captive portal must be compatible with our manner of gathering information.
- Web server containing the captive portal and the AAA servers, which are located in external sites.

1.2.1 Subscriber sessions

A subscriber is defined as a physical device that must be authorized to access the network through a HotSpot gateway. Every subscriber has a HotSpot session characterized by its MAC address.

A subscriber session is initially created in *unauthorized* status. That means traffic coming from this device is classified as *non-accepted* in the HotSpot, and access policies are applied for that type of traffic. To change the session status to *authorized*, a process of authentication and authorization of the subscriber must be carried out using a method based on a captive portal (detailed in the following sections). When a session status switches to *authorized*, every packet from the subscriber defined is classified as *accepted* and no access policies are applied to it.

A **white list** is made up of MAC addresses belonging to subscribers whose packets are classified as *accepted* in the HotSpot, regardless of the session status. Therefore, access policies and restrictions defined for *non-accepted* packets are not applied to them.

Another tool to classify the IP packets as *accepted* is the walled garden. The **walled-garden** is a set of criteria defined at IP or an upper level (for example, in a access list). Packets that match these criteria are automatically classified as *accepted*, even if the subscriber session is *unauthorized*.

There are two ways in which access policies are applied to packets classified as *non-accepted*:

(a) If it is an HTTP packet, it is intercepted and a NAT sets the internal HTTP server embedded in the router as new destination. This way, the router supplants the original destination and responds with an HTTP message that redirects the subscriber to the captive portal URL.

(b) If it is not an HTTP packet, it is discarded or labeled for further processing (depending on the administrator configuration).

A client is defined as a source of IP traffic (a source IP address). This means a subscriber could have one or more IP clients in the HotSpot database. When a session associated to a MAC address is *authorized*, all traffic sent by IP clients from said MAC address is classified as *accepted*.

A session switches from *authorized* to *unauthorized* if any of the following events occurs:

- Session lifetime is exceeded.
- Session time without activity is exceeded.
- The traffic quota allowed (received or transmitted) is exceeded.
- An explicit request from the subscriber is dealt with (see 1.2.3.1 section for more information).
- An explicit request from the administrator is dealt with (see 1.2.4 section for more information).

Every *authorized* session has a set of attributes whose value can be assigned from the defaults configured in the HotSpot feature, or that can be sent for that specific session by an AAA server. Any value received in an AAA message takes preference over the one configured in the HotSpot. The attributes defined for a subscriber session are the following:

- **Session timeout**: time (in seconds) a session remains in *authorized* status. After the session timeout has expired, the session becomes *unauthorized*. The default value is 3600 seconds (1 hour).
- **Idle timeout (optional)**: time (in seconds) a session remains in *authorized* status without subscriber activity. After session idle timeout has expired, the session becomes *unauthorized*.
- **Uplink bandwidth limit (optional)**: bitrate (kbps) limit when it comes to receiving IP traffic from a subscriber in *authorized* status.
- **Downlink bandwidth limit (optional)**: bitrate (kbps) limit for transmitting IP traffic to a subscriber in *authorized* status.
- **Uplink burst committed (optional)**: maximum burst size (bytes) allowed if bandwidth limit is configured.
- **Downlink burst committed (optional)**: maximum burst size (bytes) allowed if bandwidth limit is configured.
- **Uplink burst excess (optional)**: burst excess (bytes) allowed if bandwidth limit is configured.
- **Downlink burst excess (optional)**: burst excess (bytes) allowed if bandwidth limit is configured.
- **Uplink maximum octets (optional)**: maximum number of bytes the HotSpot can receive from a subscriber in *authorized* status. After exceeding this quota, the session changes to *unauthorized* status.
- **Downlink maximum octets (optional)**: maximum number of bytes the HotSpot can transmit to a subscriber in *authorized* status. After exceeding this quota, the session changes to *unauthorized* status.

- **Uplink queue length (optional)**: maximum packet queue size (packets) if bandwidth limit is configured.
- **Downlink queue length (optional)**: maximum packet queue size (packets) if bandwidth limit is configured.
- **Accounting interim interval (optional)**: time interval (in seconds) to send periodic accounting records of sessions in *authorized* status.
- **Filter-Id (optional)**: name of an access list pre-configured in the router. Said list contains a set of IP or upper level criteria and acts as a filter, discarding non-matching traffic. This attribute is used in both *authorized* and *unauthorized* sessions. For *unauthorized* sessions, the filter is applied after the walled garden.
- **Redir-URL (optional)**: this attribute is used only in *unauthorized* sessions. It is the captive portal URL to which this specific subscriber must be redirected.
- **User-URL (optional)**: this attribute is used only in *unauthorized* sessions. It is the destination URL to which a subscriber must be redirected if the authentication process works using the UAM server (see section 1.2.3 for more information).

1.2.2 Redirecting HTTP traffic to the Captive Portal

Received HTTP traffic classified as *non-accepted* is intercepted by the HotSpot and a destination NAT (DNAT) sets an embedded HTTP server located in the router as new destination. This server builds an HTTP response, redirecting the subscriber's HTTP client to the URL of the captive portal. The embedded server is called the Universal Authentication Method (UAM) server and is described further on.

In this HTTP redirection, the new destination URL is built by adding a set of parameters to the configured portal URL. These are used to provide information about the client and the router to the captive portal. In the router, the URL portal value can be set in two different ways, with the following priority order:

(1) URL set in the specific attribute used for this matter in this subscriber session. The value of this attribute can be received in the response from an AAA server.

(2) URL configured through the **portal-page** command.

In the second case, when the HotSpot redirects to the URL configured by the **portal-page** command, the following parameters are added to build the URL:

Parameter	Description
uamip	IP address in which the UAM server is listening (see section 1.2.3 for more information).
uamport	TCP port in which the UAM server is listening (see section 1.2.3 for more information).
uamportssl	TCP port in which the UAM server is listening for HTTPS requests (see section 1.2.3 for more information).
challenge	Temporal challenge generated for the client used in CHAP authentication. It is only added if CHAP authentication is enabled in the HotSpot.
called	MAC address used to identify the router.
mac	MAC address of the subscriber device.
nasid	Optional. NAS identifier given to the router.
ip	IP address of the client.
userurl	Original URL requested by the client.
status	Session status represented by a integer value 1 (for *authorized*) or 0 (for *unauthorized*).
coaip	Optional. IP address where the router is listening for CoA commands (see section 1.2.4 for more information).
coaport	Optional. UDP port where the router is listening for CoA commands (see section 1.2.4 for more information).
md	This message digest is included only if a shared secret between the router and the captive portal is configured. It is the MD5 hash calculated over all previous parameters.

The URL portal value received from an AAA server is built only by adding parameters *userurl*, *challenge*, *md*, and all the configured optionals.

Finally, the URL syntax is as follows:

```
http[s]://<portal-page>?param1=value1&param2=value2&param3=value3...
```

String parameter values (e.g., *userurl*) are added to the encoded URL.

Example of URL built when the value configured by the **`portal-page`** command is used:

http://any-domain.com/index.html?uamip=192.168.1.1&uamport=4532&called=08-a9-d2-6b-76-42&mac=08-a1-d2-23-76-88&ip=192.168.1.15&userurl=www.google.com&status=0

All these parameters can be parsed and used by the captive portal to generate customized web pages for a given client, update roaming databases, dynamically create users in AAA servers, etc.

1.2.3 The UAM Server

The Universal Authentication Method (UAM) is a browser-based login method where a client provides user credentials through HTTP requests. Authentication is then carried out at the network and transport layers. In our proprietary solution, a captive portal mechanism is implemented for this task.

Wireless ISP Roaming 2.0 (WISPr 2.0) specifications, published by the Wireless Broadband Alliance (WBA), include a Universal Authentication Method, widely taken as reference, to implement login protocols using captive portals. Our solution is based on the WISPr 2.0 recommendations described in said document.

The UAM server is an embedded HTTP server located in the router. It is in charge of receiving HTTP messages from clients and performing some kind of action over their subscriber sessions (authenticate, de-authenticate, ...).

By enabling HTTP redirection in the HotSpot feature, *non-accepted* HTTP traffic is received by the internal UAM server. The latter builds an HTTP response, redirecting the client to a captive portal page. The following diagram shows a basic sequence of events for user authentication:

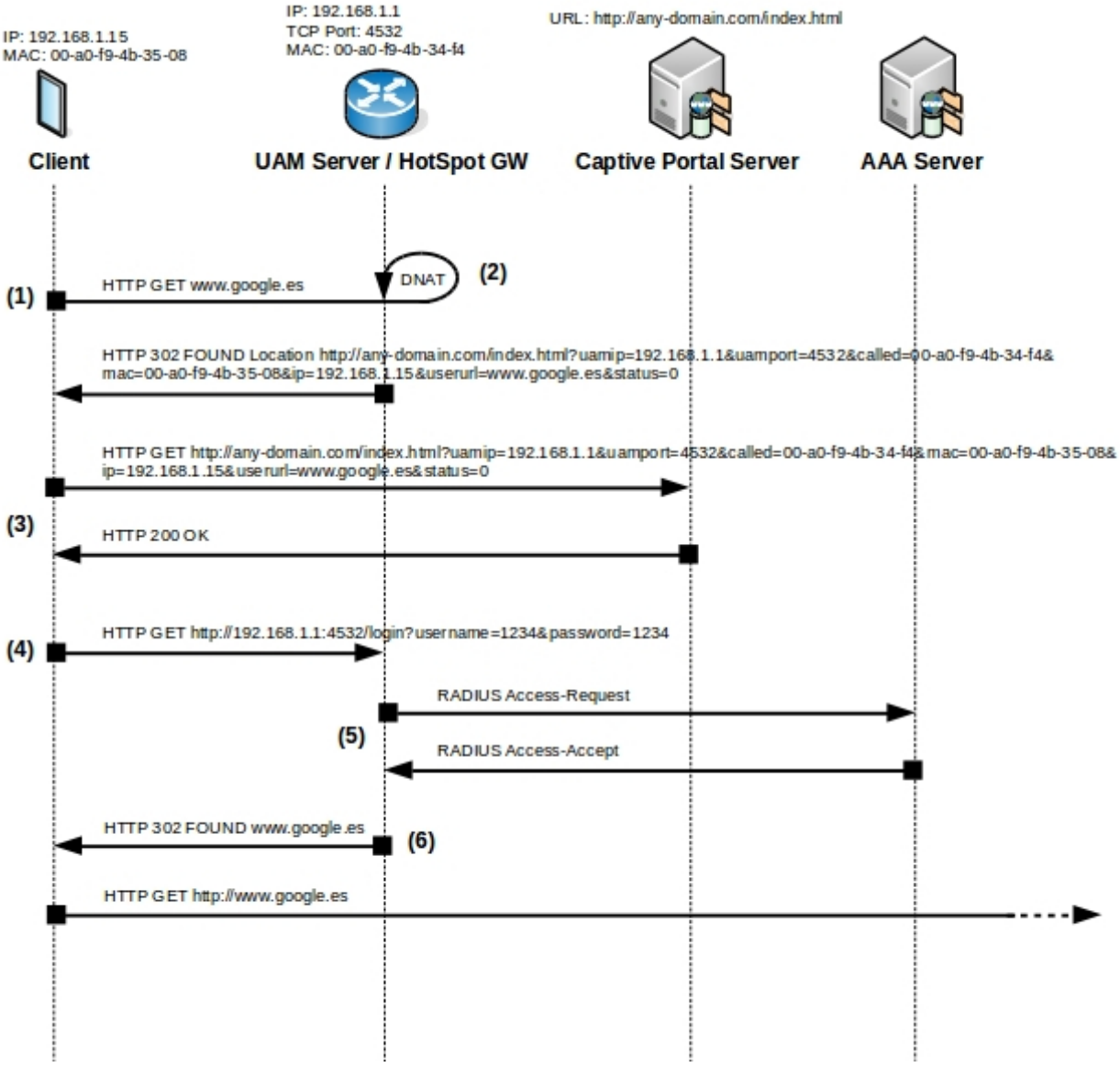

Fig. 2: **Traffic flow diagram. HotSpot GW solution with UAM Server authentication.**

(1) A client starts a web browser and requests an arbitrary URL. A subscriber session is created in the HotSpot for the client in *unauthorized* status.

(2) The HotSpot intercepts the HTTP request, classifying it as *non-accepted* traffic. The UAM server receives the request and redirects the client to the portal page. Non-HTTP traffic classified as *non-accepted* is dropped or labeled (depending on the policy configured).

(3) The captive portal page is presented to the client.

(4) Some communication between the captive portal and the client takes place. Eventually, an HTTP request from

the client must be sent to the UAM server for user credentials submission.

(5) The HotSpot sends an Access-Request to the AAA server using said credentials. The AAA server then responds with an accept/reject message, which might include session attributes such as session timeout, idle timeout, etc.

(6) After successful authentication, the subscriber session switches to *authorized* status and the client is redirected by HTTP to the URL originally requested. Optionally, administrators can redirect clients to a specific URL based on the authentication result.

1.2.3.1 HTTP messages received by the UAM Server

HTTP messages received from clients can be handled in two different ways:

(a) HTTP requests for an arbitrary URL, originally destined to some remote HTTP server, may be intercepted by the HotSpot feature and redirected to the UAM server by means of a destination NAT.

(b) HTTP requests can be directly destined to the UAM server to make a query or perform some change over the subscriber session.

Changes in the subscriber session linked to HTTP requests directly destined to the UAM server are defined by the URL path. The syntax of the URL must be one of the following:

- URLs that trigger an authentication process:

```
http[s]://<uamip>:[<uamport>]/login
```

```
http[s]://<uamip>:[<uamport>]/logon
```

- URLs that trigger a deauthentication process:

```
http[s]://<uamip>:[<uamport>]/logout
```

```
http[s]://<uamip>:[<uamport>]/logoff
```

- URLs that request information on the subscriber session:

```
http[s]://<uamip>:[<uamport>]/status
```

- Special URLs whose authentication is compatible with Bintec's captive portal:

```
http[s]://<uamip>:[<uamport>]/auth?action=login
```

After an authentication process, and depending on its result, a HTTP message redirects the client to an URL page. If the authentication is successful, the HTTP response redirects the client to an URL page according to the following priority order:

(1) Value set in the *userurl* attribute of the subscriber session. This attribute can be received in the HTTP request, together with the user credentials (see section 1.2.3.2 for more information), or in a response from an AAA server.

(2) URL configured through the **`success-page`** command.

(3) Original URL requested by the client and remembered by the router.

If the authentication process fails, the HTTP response redirects the client to an URL page according to the following priority order:

(1) URL configured through the **`fail-page`** command.

(2) Captive portal URL.

1.2.3.2 Receiving user credentials through HTTP requests

To obtain the user credential, the client must send an HTTP message to the UAM server requesting one of the described URLs used to trigger an authentication process (see section 1.2.3.1 for more information). This HTTP message contains a set of parameters where credentials are specified.

In this HTTP request, the method used must be of type GET or POST. If the method is GET, the parameters with credentials must be added to the URL following a URL query format:

```
http[s]://<uamip>:[<uamport>]/login?param1=value1&param2=value2&...
```

Where *param1*, *param2*, etc. are the credentials, and *value1*, *value2*, etc. are their respective values.

If the HTTP method is POST, the fields with credentials must be included in the body of the HTTP message as indicated here:

```
param1=value1&param2=value2&...
```

The following table lists predefined parameters the UAM server can catch:

Parameter	Description
username or *user*	User name used to perform authentication with the AAA server.
password or *pass*	User password used to perform authentication with the AAA server.
response	User CHAP password used to perform CHAP authentication with the AAA server.
ident	User CHAP identity used to perform CHAP authentication with the AAA server.
userurl	URL used to redirect the user if the authentication succeeds.
callback	Javascript callback name. Used if parameters are supplied using JSON.
action	Only needed to trigger a `login` action that is Bintec supported. Valid parameter value is *action=login*.

Example of how to submit credentials using an URL query string format:

http://172.16.0.254:4532/login?username=my-name&password=123456

Example of how to submit credentials, using a URL query string format, to perform CHAP authentication with the AAA server:

http://172.16.0.254:4532/login?username=my-name&response=23ad34b8af202aab04659575d3074d2b&ident=1

The *username* (or *user*) parameter is mandatory in this HTTP message. Additionally, the message must include a parameter that specifies the password (*password*, *pass* or *response*) for this user.

1.2.3.3 User authentication using the UAM Server

The HotSpot feature provides different mechanisms for user authentication. By enabling the Challenge-Handshake Authentication Protocol (CHAP) in the UAM server, the HotSpot generates a 16 bytes random challenge for every subscriber. This challenge is sent in the initial redirect response to the client that is able to start a CHAP authentication process. Additionally, a shared secret between the UAM server and the captive portal can be used to encrypt the user password. Said secret should never be sent to the network.

If a CHAP-based RADIUS authentication is desired, the shared secret between the UAM server and the captive portal must match the secret used between the RADIUS client and the RADIUS server.

The authentication process consists of three phases:

(1) A client with an *unauthorized* session tries to request some arbitrary URL, and the UAM server responds with a HTTP redirect to the captive portal. If CHAP authentication is enabled in the UAM server, a challenge for that subscriber is generated, valid for 10 minutes. This challenge is included in the HTTP response which redirects to the portal.

(2) Eventually, the client must send a HTTP request to the UAM server to supply the required username and password (ciphered or not). This password must be added with one of the following parameter identifiers: *password*, *pass* or *response* (see section 1.2.3.2 for more information).

Performing a PAP or CHAP authentication is up to the administrator. The following pseudo-code shows how to calculate the password before supplying it to the UAM server, according to the authentication type desired:

```
if "CHALLENGE" available
    newchallenge = SECRET? MD5(CHALLENGE, SECRET) : CHALLENGE
    if "PAP" desired
        password = XOR(PASSWORD, newchallenge)
        Send HTTP Login Request to UAM Server
    else if "CHAP" desired
        response = MD5(IDENT, PASSWORD, newchallenge)
        ident = IDENT
        Send HTTP Login Request to UAM Server
    end if
else
    password = PASSWORD
    Send HTTP Login Request to UAM Server
end if
```

When `CHALLENGE` and `SECRET` are both known by the portal and the UAM server, `password`, `response`, `ident` and `newchallenge` are calculated. `PASSWORD` and `IDENT` depend on the type of value entered by the user on the login form.

(3) In the HTTP request received by the UAM server, if the password is supplied using the *password* or *pass* parameters, an authentication process of type PAP is triggered. On the other hand, if the password is supplied in the *response* parameter (along with the *ident* parameter, which is assumed to be zero unless otherwise specified), a CHAP-based authentication process is triggered.

The following pseudo-code shows how the UAM server calculates the RADIUS attributes used to perform the authentication on the AAA server:

```
if "CHALLENGE" available
    newchallenge = SECRET? MD5(CHALLENGE, SECRET) : CHALLENGE
    if "password" received
        User-Password = XOR(password, newchallenge)
        Send PAP RADIUS request
    else if "response" received
        CHAP-Password = ident + response
        CHAP-Challenge = newchallenge
        Send CHAP RADIUS request
    end if
else if "password" received
    User-Password = password
    Send PAP RADIUS request
end if
```

When `CHALLENGE` and `SECRET` are both known by the portal and the UAM server, `password`, `response` and `ident` are received in the HTTP request and `newchallenge`, `User-Password`, `CHAP-Password` and `CHAP-Challenge` are calculated.

1.2.4 MAB and CoA functionalities

Using the MAC Authentication Bypass (MAB) and the Change of Authorization (CoA) functionalities jointly in the HotSpot provides an alternative to the UAM server where user authentication based on captive portal is desired. These two tools provide an easy solution (from the user's point of view) in situations where there is roaming between different HotSpots that share the same database in an AAA server, since redirection to the captive portal is no longer needed if the subscriber has previously authenticated a session in the AAA server of a different HotSpot.

The MAC Authentication Bypass (MAB) feature is used to request access at the link layer for traffic from a device in the interface where the HotSpot is configured. This access is resolved for a given source MAC address and, according to the result, the traffic is discarded or accepted to be processed further by the router.

On the other hand, the Change of Authorization (CoA) feature allows the administrator to dynamically perform changes over a subscriber session. Changes are requested by a remote CoA client using some AAA protocol (nowadays only RADIUS is available), and the router responds by accepting or rejecting the request. The commands provided by the CoA feature to perform actions over the sessions created in the HotSpot are the following:

Command	Description
Disconnect	With the reception of this command, the HotSpot ends the subscriber session. That means the session changes its status from *authorized* to *unauthorized*. If the change is successfully done, the router responds with an acknowledgment.
Hotspot:Reauthenticate	This command orders the HotSpot to issue a new MAB request for a given subscriber session. If the subscriber the CoA request points at has an existing session in the HotSpot, the router responds with an acknowledgment and issues a new MAB request to the AAA server.

By enabling the MAB feature, a new authentication process triggers in the HotSpot once the first packet from an unknown subscriber MAC address is received. The response to this MAB request from the AAA server sets the access level for the packets received from this subscriber:

(1) If the MAB request is rejected, all packets from that subscriber are discarded in the incoming interface (where the HotSpot is enabled) for a while. Afterwards, the reception of a new packet triggers a new MAB request and the process is repeated.

(2) If the MAB request is accepted, and the response from the AAA server contains an attribute designed to provide the captive portal URL, packets from the MAC address are not discarded and the client can be redirected to the portal's URL through HTTP. The subscriber session remains *unauthorized* until a CoA *Hotspot:Reauthenticate* command is received. The latter triggers a new MAB request, which again leads to one of these options.

(3) If the MAB request is accepted, and the response from the AAA server does not contain an attribute providing an URL of the captive portal, packets from the MAC address are not discarded and the session status changes from *unauthorized* to *authorized*. The AAA response also includes the attributes that are necessary to configure an *authorized* session: session timeout, idle timeout, interim interval, etc.

Compared with the UAM server, the sequence of exchanged messages is quite different when MAB and CoA functionalities are used. The following diagram shows the messages that intervene in the authentication of a subscriber that was not previously authenticated in the AAA server:

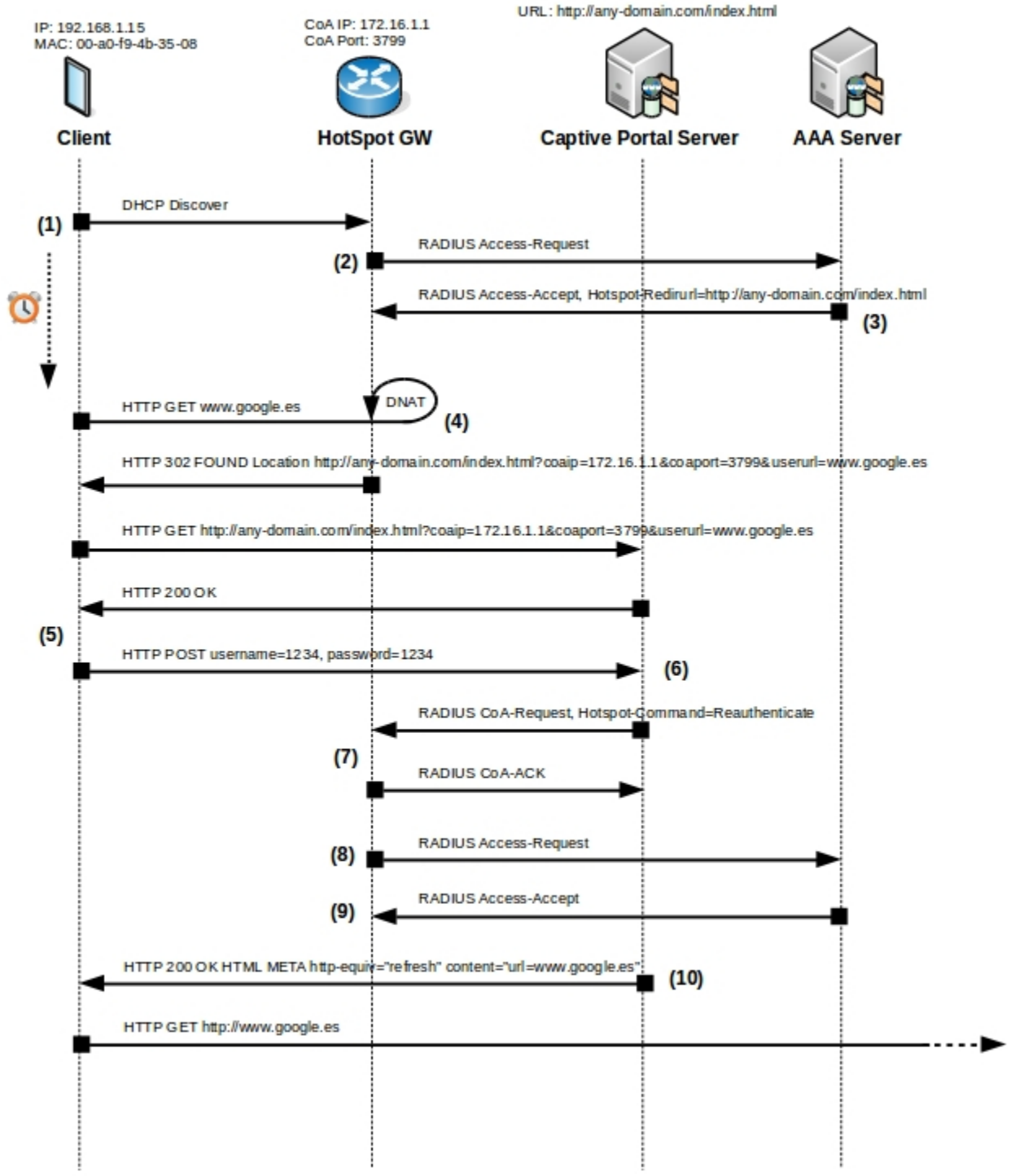

Fig. 3: **Traffic flow diagram. HotSpot GW solution with MAB and CoA authentication.**

(1) The HotSpot receives a packet from an unknown subscriber (typically the broadcast DHCP Discover packet).

(2) The HotSpot creates a new *unauthorized* session for the unknown subscriber and sends an MAB request to the AAA server for the associated MAC address.

(3) The AAA server accepts the MAB request. Since the subscriber had not yet authenticated in the AAA server, the response contains an attribute with the captive portal URL and the session remains in *unauthorized* status.

(4) The subscriber requests an arbitrary URL using an HTTP client, typically the WEB browser. The HotSpot feature intercepts the request and responds by redirecting the client to the captive portal. The remaining traffic is classified as *non-accepted* and discarded or labeled (depending on the configuration).

(5) The client receives the portal page with the login form. After filling the form, the client submits the user credentials to the captive portal.

(6) The captive portal verifies the credentials and sends a CoA command to the HotSpot. This command includes an attribute that forces the HotSpot to reauthenticate and reauthorize the subscriber MAC. A synchronization of databases between the captive portal and the AAA server is done in the background, so the AAA server knows that the subscriber provided valid credentials.

(7) The HotSpot receives the CoA command and the attributes in the message are used to identify the subscriber session. After verifying a session is created for this subscriber, the router responds with a CoA ACK.

(8) The HotSpot issues a new MAB request to the AAA server, just like the CoA command ordered.

(9) The AAA server also accepts the request, but this time the portal URL is not included in the AAA response. Consequently, the session status changes from *unauthorized* to *authorized* in the HotSpot. In this AAA response, the AAA server also includes all the attributes it wishes the session to have: session timeout, idle timeout, interim interval, etc.

(10) Based on the results, the captive portal server sends an HTTP response redirecting the client to an URL.

1.2.5 AAA services

The HotSpot feature uses Radius to implement Authentication, Authorization and Accounting services. Therefore, a Radius client must be properly configured in the equipment.

The first available option to configure a Radius client is to use the Radius protocol feature. Radius does not allow the administrator to configure subscriber sessions accounting. Please see manual *Dm733-I Radius Protocol* for further information.

A Radius client can also be configured by enabling the AAA feature. The AAA feature offers the possibility of configuring subscriber sessions accounting. Please see manual *Dm800-I AAA Feature* for further information.

1.2.5.1 Authentication and Authorization

To authenticate and authorize a subscriber session, the Radius client must send an Access-Request to a Radius server. The Radius attributes included in the Access-Request depend on the type of authentication configured. Two different authentication types are defined:

(a) Based on the user and password submitted by the subscriber. This is the only type available when the Radius client is configured by the Radius protocol feature in the router. If the AAA feature is enabled, an **authentication login** AAA method list must be configured and assigned to the HotSpot. Please see manual *Dm800-I AAA Feature* for further information.

(b) Based on the MAC address of the subscriber. This is used when the MAC Authentication Bypass (MAB) is enabled in the HotSpot. This authentication type is only available if the AAA feature is enabled. An **authentication dot1x** AAA method list must be configured and assigned to the HotSpot. Please see manual *Dm800-I AAA Feature* for further information.

The attributes included in a RADIUS Access-Request are the following:

Attribute	Description
User-Name	User name submitted by the subscriber in case of authentication based on the user and password, or MAC address if the authentication is based on the MAC address.
User-Password	Password submitted by the subscriber in case of authentication based on the user and password, or ciphered MAC address if the authentication is based on the MAC address.
Chap-Password	User password for CHAP authentication if the UAM server is used.
Chap-Challenge	User challenge for CHAP authentication if the UAM server is used.
Nas-Ip-Address	The IP address of the RADIUS client in the router.
Nas-Identifier	Optional. The NAS identifier of the RADIUS client in the router.
Service-Type	Text value defined in the IETF RFC 2865. The text `LOGIN` is sent when the UAM server is used. The text `CALL-CHECK` is sent when the MAB feature is used.
Calling-Station-Id	Subscriber device MAC address.
Called-Station-Id	Network interface MAC address where the HotSpot is configured.
Framed-Ip-Address	Client IP address.
Nas-Port-Type	Interface type where the HotSpot is configured. Valid values are defined in IETF RFC 2865.
Nas-Port	Interface number where the HotSpot is configured.
Acct-Session-Id	Optional. Unique ID assigned to the accounting session.

In a Radius Access-Accept response sent by a Radius server, two types of authorization attributes can be applied to a subscriber session: standard Radius attributes and vendor-specific attributes.

Standard Radius attributes are defined in IETF RFC 2865. The list of standard attributes that can be applied to a subscriber session is as follows:

Attribute	Description
Session-Timeout	Time (in seconds) a session remains in *authorized* status. After session timeout has expired, the session must be reauthorized.

Idle-Timeout	Time (in seconds) a session remains in *authorized* status without subscriber activity. After a session idle timeout has expired, the session must be reauthorized.
Class	Unmodified string to be sent in accounting records. This must not be interpreted by the Radius client.
Acct-Interim-Interval	Time interval (in seconds) for sending periodic accounting records for sessions in *authorized* status.
Filter-Id	String value to identify a pre-configured access list in the router. This list defines a set of criteria at the IP or upper level. Acting as a filter, it discards the traffic that does not match these criteria.

To offer a more customizable HotSpot feature, some proprietary Radius attributes have been designed to be interpreted and used in the session authorization process. These proprietary attributes must be sent with the proper vendor-specific identifier.

The list of vendor-specific attributes that can be applied to a subscriber session is as follows:

Attribute	Vendor-Type	Description
Bandwidth-Kbps-Up	4	Integer used to define the bitrate (kbps) limit for receiving IP traffic from an *authorized* subscriber.
Bandwidth-Kbps-Down	5	Integer used to define the bitrate (kbps) limit for transmitting IP traffic to an *authorized* subscriber.
Bandwidth-Bc-Up	6	Integer used to define the burst size (bytes) allowed when a bandwidth limit is configured.
Bandwidth-Bc-Down	7	Integer used to define the burst size (bytes) allowed when a bandwidth limit is configured.
Bandwidth-Be-Up	8	Integer used to define the burst excess (bytes) allowed when a bandwidth limit is configured.
Bandwidth-Be-Down	9	Integer used to define the burst excess (bytes) allowed when a bandwidth limit is configured.
Max-Octets-Up	10	Integer used to set the maximum number of bytes the HotSpot can receive from an *authorized* subscriber.
Max-Octets-Down	11	Integer used to set the maximum number of bytes the HotSpot can transmit to an *authorized* subscriber.
Queue-length-Up	12	Integer used to define the packet queue size (number of packets).
Queue-length-Down	13	Integer used to define the packet queue size (number of packets).
Hotspot-Userurl	15	String with the redirect URL after successful authentication with the UAM server for a specific subscriber.
Hotspot-Redirurl	18	String with the URL of the captive portal for a specific subscriber.

1.2.5.2 Accounting

Subscriber session accounting means enabling the AAA feature. Accounting records can be sent at the beginning and the end of a session and, optionally, in periodic intervals, provided the *interim-interval* parameter was preconfigured. Standard attributes for Radius accounting are defined in IETF RFC 2865 and IETF RFC 2866.

The list of attributes included in an accounting record is as follows:

Attribute	Description
User-Name	User name submitted by the subscriber in case of authentication based on the user and password, or MAC address if the authentication is based on the MAC address.
Acct-Session-Id	Unique ID assigned to the accounting session.
Acct-Status-Type	Type of accounting record: START, STOP or INTERIM.
Class	String received in the RADIUS Access-Accept response.
Nas-Ip-Address	Source network interface IP address for the accounting record.
Acct-Input-Octets	Number of bytes received from a subscriber. Only transmitted in STOP and INTERIM records.
Acct-Output-Octets	Number of bytes transmitted to a subscriber. Only transmitted in STOP and INTERIM records.

Acct-Session-Time	Time (in seconds) the subscriber session has been in *authorized* status. Only transmitted in STOP and INTERIM records.
Acct-Terminate-Cause	Session end cause. Only transmitted in STOP records.
Nas-Ip-Address	The IP address of the RADIUS client in the router.
Nas-Identifier	Optional. The NAS identifier of the RADIUS client in the router.
Framed-Ip-Address	Client IP address.
Calling-Station-Id	Subscriber device MAC address.
Called-Station-Id	Network interface MAC address where the HotSpot is configured.
Nas-Port-Type	Interface type where the HotSpot is configured. Valid values are defined in IETF RFC 2865.
Nas-Port	Interface number where the HotSpot is configured.
Acct-Authentic	Optional. Protocol used for authentication. Valid values are defined in IETF RFC 2865.
Event-Timestamp	Optional. Time instant the accounting record was generated, in seconds, since January 1, 1970 00:00 UTC.

1.2.5.3 Change of Authorization

The Change of Authorization (CoA) feature provides a mechanism to perform occasional changes on a subscriber session when the router receives certain RADIUS messages. To do this, the protocol extension standard for RADIUS Dynamic Authorization is defined in IETF RFC 5176.

The use of RADIUS Dynamic Authorization in the router is obtained by configuring the Change of Authorization feature in the AAA feature. Please see manual *Dm800-I AAA Feature* for further information.

To change a subscriber session, the administrator must send some of the Dynamic Authorization RADIUS messages defined in the standard to the router:

(a) **Disconnect-Request**: this message ends a subscriber session. This means the session status changes from *authorized* to *unauthorized*. If the change is successfully completed, the router responds with a Disconnect-ACK message. Otherwise, the router sends a Disconnect-NACK response, including the RADIUS attribute *Error-Cause* with the error code. Please see manual *Dm800-I AAA Feature* for further information on error codes.

(b) **CoA-Request**: this message is used to notify the router that some change on the session authorization must be done. The specific change to perform is given by the RADIUS attributes included in the request. If the change is successfully completed, the router responds with a CoA-ACK message. Otherwise, the router sends a CoA-NACK response, including the RADIUS attribute *Error-Cause* with the error code. Please see manual *Dm800-I AAA Feature* for further information on error codes.

Any Dynamic Authorization request must contain one or several RADIUS attributes used to identify the subscriber session in the router. The following attributes allow the HotSpot to identify a session:

- Acct-Session-Id (attribute #44)
- Calling-Station-Id (attribute #31)
- User-Name (attribute #1)

A session is only considered identified if all the attributes listed above and included in the request match the ones in the session.

In a CoA-Request, the RADIUS attributes allow for a subscriber session to be modified. The vendor-specific attributes provided for this purpose are the following:

Attribute	Vendor-Type	Description
Hotspot-Command	14	Integer that identifies the change to perform over a subscriber session in the HotSpot. This attribute can only be received in a CoA-Request.

According to the *Hotspot-Command* attribute value, the following actions can be validly carried out:

Command	Value	Description
Hotspot:Reauthenticate	0	This command forces the HotSpot to issue a new MAB request for a given subscriber session. If the subscriber has created a session in the HotSpot, the router responds with a CoA-ACK and sends a new MAB request to the AAA server.

Chapter 2 Configuration

2.1 Global Configuration

To access the Hotspot configuration menu, use the **feature hotspot** command found in the main configuration menu.

Example:

```
Config>feature hotspot

-- Hotspot Configuration --
HS config>
```

In the HotSpot feature's global configuration menu, the following commands are available:

Command	Function
? (HELP)	Displays the configuration commands or their options.
DEBUG	Debugging options of the Hotspot feature.
NETWORK	Configures a Hotspot on a network interface.
NO	Configures parameters with their default values.
EXIT	Exits the configuration menu.

2.1.1 ? (HELP)

Displays the commands available and their options.

Command history:

Release	Modification
11.00.03	The "*? (Help)*" command was introduced as of version 11.00.03.

2.1.2 DEBUG

Enables additional printing of HotSpot subsystem events for debugging purposes.

Syntax:

```
HS config>debug enable
```

To disable additional printing of HotSpot subsystem events, enter **no debug**.

Command history:

Release	Modification
11.00.03	The "*debug*" command was introduced as of version 11.00.03.

2.1.3 NETWORK

Accesses the HotSpot configuration menu for a specific interface. The HotSpot feature can be applied to the following:

- Ethernet interfaces (type `ethernetX/Y`).
- Ethernet subinterfaces (type `ethernetX/Y.Z`).
- WLAN interfaces (type `wlanX/Y`).
- WLAN subinterfaces (type `wlanX/Y.Z`).
- BVI interfaces (type `bviX`).
- BVI subinterfaces (type `bviX.Z`).

Syntax:

```
HS config>network <Interface name>
```

Example:

```
HS config>network wlan0/0


Network wlan0/0>
```

To delete a HotSpot configured in an interface, enter **no network <Interface name>**.

Note

Any changes performed in dynamic configuration mode require the HotSpot feature to be disabled and re-enabled in the network interface.

Command history:

Release	Modification
11.00.03	The "*network*" command was introduced as of version 11.00.03.
11.00.03, 11.01.00	The HotSpot function can be applied to Ethernet and WLAN subinterfaces.
11.00.05, 11.01.01	The HotSpot function can be applied to BVI interfaces and subinterfaces.

2.1.4 NO

Configures parameters and their default values.

Syntax:

```
HS config>no ?
  debug      Debugging options for Hotspot feature
  network    Configure a hotspot network
```

Command history:

Release	Modification
11.00.03	The "*no*" command was introduced as of version 11.00.03.

2.1.5 EXIT

Exits the HotSpot configuration menu.

Command history:

Release	Modification
11.00.03	The "*exit*" command was introduced as of version 11.00.03.

2.2 Network Configuration

Configures HotSpot for a specific interface. The HotSpot feature can be applied to the following network interfaces:

- Ethernet interfaces (type `ethernetX/Y`).
- Ethernet subinterfaces (type `ethernetX/Y.Z`).
- WLAN interfaces (type `wlanX/Y`).
- WLAN subinterfaces (type `wlanX/Y.Z`).
- BVI interfaces (type `bviX`).
- BVI subinterfaces (type `bviX.Z`).

The following table shows commands in the network configuration menu:

Command	Function
? (HELP)	Displays the configuration commands or their options.
ACCOUNTING	Configures a list of AAA methods for accounting.
ACL-ALIAS	Configures an alias to identify an access list.

AUTHENTICATION	Configures a list of AAA methods for authentication.
AUTHORIZATION	Configures a list of AAA methods for authorization.
CHANGE-OF-AUTHORIZATION	Configures Change of Authorization in subscriber sessions.
CLIENTS-FILTER	Configures a filter to limit access to invalid IP addresses.
ENABLE	Enables HotSpot in the interface.
MAC-AUTHENTICATION-BYPASS	Configures MAC Authentication Bypass.
MAX-SESSIONS	Maximum number of sessions connected simultaneously.
NO	Configures parameters with their default values.
POLICY	Access policy applied to *non-accepted* packets received from unauthenticated subscribers.
REDIRECT	Redirects HTTP traffic to the UAM server for unauthenticated subscribers.
SESSION	Configures default values for session parameters.
STATUS-API-REQUEST	Configures API/HotSpot parameters.
UAM-SERVER	Configures the UAM server.
URL	Configures URLs for HTTP redirects.
WALLED-GARDEN	Defines the IP traffic that is always accepted (even for unauthenticated subscribers).
WHITE-LIST	Configures the list of subscribers that do not require authentication.
EXIT	Exits the configuration menu.

Even though the HotSpot feature can be configured in several networks, from here onwards `wlan0/0` will be used in the examples and syntax rules included in the following sections.

Note

Any changes made in dynamic configuration mode require the HotSpot feature to be disabled and re-enabled in the network interface.

2.2.1 ? (HELP)

Displays the commands available and their options.

Command history:

Release	Modification
11.00.03	The "*? (Help)*" command was introduced as of version 11.00.03.

2.2.2 ACCOUNTING

Configures HotSpot accounting.

Syntax:

```
Network wlan0/0>accounting {interim-interval <value> | network <AAA list name>}
```


Note

Accounting is only available when the AAA feature is enabled.

Command history:

Release	Modification
11.00.03	The "*accounting*" command was introduced as of version 11.00.03.

2.2.2.1 ACCOUNTING INTERIM-INTERVAL

Configures an interval to send periodic INTERIM accounting records. To enable the sending of INTERIM records, `start-stop` actions must be configured in the list of AAA methods that enable session accounting. Please see manual *Dm800-I AAA Feature* for further information on AAA configuration.

Syntax:

```
Network wlan0/0>accounting interim-interval <value>
```

Example:

```
Network wlan0/0>accounting interim-interval 10m
```

To disable periodic accounting records sending, enter **no accounting interim-interval**.

Note

Accounting is only available when the AAA feature is enabled.

Command history:

Release	Modification
11.00.03	The "*accounting interim-interval*" command was introduced as of version 11.00.03.

2.2.2.2 ACCOUNTING NETWORK

Configures a list of accounting-related AAA methods in a HotSpot. Please see manual *Dm800-I AAA Feature* for further information.

Syntax:

```
Network wlan0/0>accounting network <AAA list name>
```

Example:

```
Network wlan0/0>accounting network my-list
```

To delete a list of accounting-related AAA methods, enter **no accounting network**.

Note

Accounting is only available when the AAA feature is enabled.

Command history:

Release	Modification
11.00.03	The "*accounting network*" command was introduced as of version 11.00.03.

2.2.3 ACL-ALIAS

Configures an alias for a pre-configured access list in the router. This alias is used to identify an access list when Radius attribute *Filter-Id* is received in a session authorization.

Syntax:

```
Network wlan0/0>acl-alias <text> list <ACL number>
```

Example:

```
Network wlan0/0>acl-alias POST_AUTH_FILTER list 101
```

To delete a configured access list alias, enter **no acl-alias <text>**.

Command history:

Release	Modification
11.00.06, 11.01.02	The "*acl-alias*" command was introduced.

2.2.4 AUTHENTICATION

Configures a list of authentication-related AAA methods in a HotSpot. Please see manual *Dm800-I AAA Feature* for further information.

Syntax:

```
Network wlan0/0>authentication ?
  dot1x    AAA methods list for dot1x authentication
  login    AAA methods list for login authentication
```

- Use the **authentication dot1x** command for authentication based on a MAC address. This is the method used when the MAC Authentication Bypass (MAB) is required.
- Use the **authentication login** command for authentication based on a user name and password. This is the method used when the UAM server is required.

Command history:

Release	Modification
11.00.03	The "*authentication*" command has been introduced as of version 11.00.03.
11.00.06, 11.01.02	Command option **dot1x** has been added.

2.2.4.1 AUTHENTICATION DOT1X

Configures a list of AAA methods to perform subscriber authentication based on the MAC address.

Syntax:

```
Network wlan0/0>authentication dot1x <AAA list name>
```

Example:

```
Network wlan0/0>authentication dot1x my-list
```

To delete a configured list of AAA methods, enter **no authentication dot1x**.

Command history:

Release	Modification
11.00.06, 11.01.02	The "*authentication dot1x*" command was introduced.

2.2.4.2 AUTHENTICATION LOGIN

Configures a list of AAA methods to perform subscriber authentication based on the user name and password.

Syntax:

```
Network wlan0/0>authentication login <AAA list name>
```

Example:

```
Network wlan0/0>authentication login my-list
```

To delete a configured list of AAA methods, enter **no authentication login**.

Command history:

Release	Modification
11.00.03	The "*authentication login*" command has been introduced as of version 11.00.03.

2.2.5 AUTHORIZATION

Configures a list of authorization-related AAA methods in a HotSpot. Please see manual *Dm800-I AAA Feature* for further information.

Syntax:

```
Network wlan0/0>authorization network <AAA list name>
```

Example:

```
Network wlan0/0>authorization network my-list
```

To delete a list of AAA methods for authorization, enter **no authorization network**.

Command history:

Release	Modification
11.00.03	The "*authorization*" command has been introduced as of version 11.00.03.

2.2.6 CHANGE-OF-AUTHORIZATION

Configures the Change of Authorization (CoA) feature in the HotSpot.

Syntax:

```
Network wlan0/0>change-of-authorization enable
```

To disable the CoA feature in the HotSpot, enter **no change-of-authorization**.

Note

Change of Authorization is only available when the AAA feature is enabled.

Command history:

Release	Modification
11.00.06, 11.01.02	The "*change-of-authorization*" command was introduced.

2.2.7 CLIENTS-FILTER

Configures client-filtering policies that define whether a client is valid or invalid for the HotSpot feature according to their IP address. Incoming traffic from an IP address that belongs to an invalid client is automatically discarded, regardless of the walled garden or white list configuration.

Syntax:

```
Network wlan0/0>clients-filter ?
  dhcp       Configure a filter based on the DHCP assignation
  network    IP addresses must belong to the local network
```

Multiple filters can be configured simultaneously. In this case, clients are only considered valid when they comply with each of the filters independently.

Command history:

Release	Modification
11.00.07, 11.01.02	The "*clients-filter*" command was introduced.

2.2.7.1 CLIENTS-FILTER DHCP

Configures an IP client filter based on DHCP allocation.

Syntax:

```
Network wlan0/0>clients-filter dhcp ?
  check-pools    IP addresses configured in pools of the local DHCP server are
                 required to be assigned
  required       All IP addresses are required to be assigned
```

This command displays two DHCP filters that cannot be configured simultaneously. By using the **check-pools** option, the pools of addresses defined in the local DHCP server are checked when a newly discovered client is found. The client is only considered valid if, according to the DHCP database, he is the owner of the IP address. The client is also considered to be valid if the IP address is not found in a DHCP pool.

When using the **required** option, however, all newly discovered clients must have received their IP addresses via DHCP to be considered valid. If not, they are discarded.

To remove this filter, enter **no clients-filter dhcp**.

Command history:

Release	Modification
11.00.07, 11.01.02	The "*clients-filter dhcp*" command was introduced.

2.2.7.2 CLIENTS-FILTER NETWORK

Only grants access to clients whose IP addresses belong to the local network. Since this filter is always enabled by default, a command to disable it is provided.

Syntax:

```
Network wlan0/0>clients-filter network disable
```

To enable the default network filter, enter **no clients-filter network disable**.

Command history:

Release	Modification
11.00.07, 11.01.02	The "*clients-filter network*" command was introduced.

2.2.8 ENABLE

Enables HotSpot in the interface.

Syntax:

```
Network wlan0/0>enable
```

To disable the HotSpot feature in the interface, enter **no enable**.

Command history:

Release	Modification
11.00.03	The "*enable*" command was introduced as of version 11.00.03.

2.2.9 MAC-AUTHENTICATION-BYPASS

Configures the MAC Authentication Bypass (MAB) feature in HotSpot.

Syntax:

```
Network wlan0/0>mac-authentication-bypass ?
  activity-timeout    Configure a timeout to retry a new authentication
  enable              Enable MAC Authentication Bypass
```

Note

MAC Authentication Bypass is only available when the AAA feature is enabled.

Command history:

Release	Modification
11.00.06, 11.01.02	The "*mac-authentication-bypass*" command was introduced.

2.2.9.1 MAC-AUTHENTICATION-BYPASS ACTIVITY-TIMEOUT

Configures the minimum time before trying to re-issue a new MAB request if the last one was rejected. This is known as activity timeout and its default value is 10 minutes.

Syntax:

```
Network wlan0/0>mac-authentication-bypass activity-timeout <value>
```

Example:

```
Network wlan0/0>mac-authentication-bypass activity-timeout 5m
```

To set the default activity timeout, enter **no mac-authentication-bypass activity-timeout**.

Command history:

Release	Modification
11.00.06, 11.01.02	The "*mac-authentication-bypass activity-timeout*" command was introduced.

2.2.9.2 MAC-AUTHENTICATION-BYPASS ENABLE

Enables MAC Authentication Bypass (MAB) in the HotSpot.

Syntax:

```
Network wlan0/0>mac-authentication-bypass enable
```

To disable the MAB feature in the HotSpot, enter **no mac-authentication-bypass enable**

Command history:

Release	Modification
11.00.06, 11.01.02	The "*mac-authentication-bypass enable*" command was introduced.

2.2.10 MAX-SESSIONS

Configures the maximum number of concurrent sessions that can be authenticated in HotSpot.

Syntax:

```
Network wlan0/0>max-sessions <value>
```

Example:

```
Network wlan0/0>max-sessions 5000
```

To delete the maximum number of concurrent sessions that can be authenticated in HotSpot, enter **no max-sessions**.

Command history:

Release	Modification
11.00.03	The "*max-sessions*" command was introduced as of version 11.00.03.

2.2.11 NO

Configures parameters with their default values.

Syntax:

```
Network wlan0/0>no ?
  accounting                   Configures a list of AAA methods for accounting
  acl-alias                    Configures an alias to identify an access list
  authentication               Configures a list of AAA methods for authentication
  authorization                Configure a list of AAA methods for authorization
  change-of-authorization      Configures Change of Authorization in subscriber sessions
  clients-filter               Configures a filter to limit access to invalid IP addresses
  enable                       Enables HotSpot in the interface
  mac-authentication-bypass    Configures MAC Authentication Bypass
  max-sessions                 Maximum number of sessions connected simultaneously
  policy                       Access policy applied to non-accepted packets received from unauthenticated
                               subscribers
  redirect                     Redirects HTTP traffic to the UAM server for unauthenticated subscribers
  session                      Configures default values for session parameters
  status-api-request           Configures the URL to send an HTTP/HTTPS status API-request from the HotSpot
  status-api-request-parameter Configures the API-request parameters
  uam-server                   Configures the UAM server
  url                          Configures URLs for HTTP redirects
  walled-garden                Defines the IP traffic that is always accepted (even for unauthenticated
                               subscribers)
```

```
white-list                    Configures the list of subscribers that do not require authentication
exit                          Exits the configuration menu
```

Command history:

Release	Modification
11.00.03	The "*no*" command was introduced as of version 11.00.03.

2.2.12 POLICY

Configures a HotSpot access policy to be applied to *non-accepted* packets.

Syntax:

```
Network wlan0/0>policy {drop | set label <value>}
```

Command history:

Release	Modification
11.00.03	The "*policy*" command was introduced as of version 11.00.03.

2.2.12.1 POLICY DROP

Configures a HotSpot access policy to drop *non-accepted* packets.

Syntax:

```
Network wlan0/0>policy drop
```

To delete a HotSpot access policy, enter **no policy**.

Command history:

Release	Modification
11.00.03	The "*policy drop*" command was introduced as of version 11.00.03.

2.2.12.2 POLICY SET LABEL

Configures a HotSpot access policy to set a specific label for *non-accepted* packets.

Syntax:

```
Network wlan0/0>policy set label <value>
```

Example:

```
Network wlan0/0>policy set label 99
```

To delete a HotSpot access policy, enter **no policy**.

Command history:

Release	Modification
11.00.03	The "*policy set label*" command was introduced as of version 11.00.03.

2.2.13 REDIRECT

Configures the redirection of *non-accepted* HTTP and HTTPS traffic to the internal UAM server.

Syntax:

```
Network wlan0/0>redirect enable
```

To disable the redirection of *non-accepted* HTTP and HTTPS traffic to the internal UAM server, enter **no redirect**.

Command history:

Release	Modification
11.00.03	The "*redirect*" command was introduced as of version 11.00.03.

2.2.14 SESSION

Configures default session parameters in the HotSpot. Every command applies to each subscriber session, unless the AAA access response contains the same field with a different value. If a field is not configured in the device and is not received in the AAA access response, the parameter is not applied to the new session.

Syntax:

```
Network wlan0/0>session ?
  downlink        Session downlink parameters
  hard-timeout    Subscriber session timeout regardless of activity
  idle-timeout    Subscriber session timeout if there is not activity
  uplink          Session uplink parameters
```

Command history:

Release	Modification
11.00.03	The "*session*" command was introduced as of version 11.00.03.
11.00.05, 11.01.01	The *bandwidth-down* command option has become obsolete.
11.00.05, 11.01.01	The *bandwidth-up* command option has become obsolete.
11.00.05, 11.01.01	The *max-octets-down* command option has become obsolete.
11.00.05, 11.01.01	The *max-octets-up* command option has become obsolete.
11.00.05, 11.01.01	New command option *downlink* added.
11.00.05, 11.01.01	New command option *uplink* added.

2.2.14.1 SESSION BANDWIDTH-DOWN

Configures a bandwidth limit per-session for downlink traffic in kbps.

Syntax:

```
Network wlan0/0>session bandwidth-down <value>
```

Example:

```
Network wlan0/0>session bandwidth-down 300
```

To delete a bandwidth limit per-session for downlink traffic, enter **no session bandwidth-down**.

Command history:

Release	Modification
11.00.03	The "*session bandwidth-down*" command option was introduced as of version 11.00.03.
11.00.05, 11.01.01	This command option has become obsolete.

2.2.14.2 SESSION BANDWIDTH-UP

Configures a bandwidth limit per-session for uplink traffic in kbps.

Syntax:

```
Network wlan0/0>session bandwidth-up <value>
```

Example:

```
Network wlan0/0>session bandwidth-up 128
```

To delete a bandwidth limit per-session for uplink traffic, enter **no session bandwidth-up**.

Command history:

Release	Modification
11.00.03	The "*session bandwidth-up*" command option was introduced as of version 11.00.03.
11.00.05, 11.01.01	This command option has become obsolete.

2.2.14.3 SESSION DOWNLINK

Configures downlink session parameters. The IP traffic that goes from the router to authenticated subscribers will be affected by this configuration section.

Syntax:

```
Network wlan0/0>session downlink ?
  bandwidth    Bandwidth control parameters
  qos          Quality of Service parameters
```

Command history:

Release	Modification
11.00.05, 11.01.01	The "*session downlink*" command option was introduced.

2.2.14.3.1 SESSION DOWNLINK BANDWIDTH

Shows all the configurable parameters that enable bandwidth limitation per authenticated subscriber. This feature is based on a traffic-shaping algorithm, similar to the one found in the BRS feature. For more detailed information on how it works, please see *Dm715-I BRS Feature.*

Syntax:

```
Network wlan0/0> session downlink bandwidth ?
  burst-committed    Maximum burst committed
  burst-excess       Maximum burst excess
  kbps               Maximum transmission rate
```

Command history:

Release	Modification
11.00.05, 11.01.01	The "*session downlink bandwidth*" command option was introduced.

2.2.14.3.1.1 SESSION DOWNLINK BANDWIDTH BURST-COMMITTED

Configures the maximum burst size allowed when rate limit is enabled. The value is specified in bytes. If no value is configured, this parameter will be set to an eighth part of the maximum rate. The ratio between the rate limit and the present parameter can be understood as the time interval during which the configured speed is guaranteed on average. For example, if you want to set a certain admissible rate at half-second intervals, the value configured must be half of the configured rate. The time interval must be set in the range between 7.8ms and 1s.

Syntax:

```
Network wlan0/0>session downlink bandwidth burst-committed <value>
```

Example:

```
Network wlan0/0>session downlink bandwidth burst-committed 1000
```

To delete a bandwidth limit per-session for downlink traffic, enter **no session downlink bandwidth burst-committed**.

Command history:

Release	Modification
11.00.05, 11.01.01	The "*session downlink bandwidth burst-committed*" command option was introduced.

2.2.14.3.1.2 SESSION DOWNLINK BANDWIDTH BURST-EXCESS

Configures the burst excess allowed when rate limit is enabled. The value is specified in bytes. If no value is configured, this parameter will be set to zero. This parameter adds a margin when the rate is reached,allowing the desirable rate limit to be adjusted in different scenarios.

Syntax:

```
Network wlan0/0>session downlink bandwidth burst-excess <value>
```

Example:

```
Network wlan0/0>session downlink bandwidth burst-excess 300
```

To delete a bandwidth limit per-session for downlink traffic, enter **no session downlink bandwidth burst-excess**.

Command history:

Release	Modification
11.00.05, 11.01.01	The "*session downlink bandwidth burst-excess*" command option was introduced.

2.2.14.3.1.3 SESSION DOWNLINK BANDWIDTH KBPS

Configures a bandwidth limit per session for downlink traffic in Kbps. In addition to setting the maximum rate per session, this command enables or disables the rate limit feature.

Syntax:

```
Network wlan0/0>session downlink bandwidth kbps <value>
```

Example:

```
Network wlan0/0>session downlink bandwidth kbps 2000
```

To delete a bandwidth limit per-session for downlink traffic, enter **no session downlink bandwidth kbps**.

Command history:

Release	Modification
11.00.05, 11.01.01	The "*session downlink bandwidth kpbs*" command option was introduced.

2.2.14.3.2 SESSION DOWNLINK QOS

Shows all the configurable parameters that can be used to enable QoS in each authenticated subscriber.

Syntax:

```
Network wlan0/0> session downlink qos ?
  max-octets      Maximum of total octets transmitted
  queue-length    Packet queue length
```

Command history:

Release	Modification
11.00.05, 11.01.01	The "*session downlink qos*" command option was introduced.

2.2.14.3.2.1 SESSION DOWNLINK QOS MAX-OCTETS

Configures the maximum data volume that an authenticated subscriber can receive in a session. The value is specified in bytes. If no value is configured, data volume restriction is not applied. Whenever the subscriber reaches the credit available, all the outgoing IP traffic is dropped by this feature.

Syntax:

```
Network wlan0/0>session downlink qos max-octets <value>
```

Example:

```
Network wlan0/0>session downlink qos max-octets 300000000
```

To delete a bandwidth limit per-session for downlink traffic, enter **no session downlink qos max-octets**.

Command history:

Release	Modification
11.00.05, 11.01.01	The "*session downlink qos max-octets*" command option was introduced.

2.2.14.3.2.2 SESSION DOWNLINK QOS QUEUE-LENGTH

Configures the packet queue size. The value represents the number of packets the queue can accept. If no value is configured, this parameter will be set to ten.

Syntax:

```
Network wlan0/0>session downlink qos queue-length <value>
```

Example:

```
Network wlan0/0>session downlink qos queue-length 50
```

To delete a bandwidth limit per-session for downlink traffic, enter **no session downlink qos queue-length**.

Command history:

Release	Modification
11.00.05, 11.01.01	The "*session downlink qos queue-length*" command option was introduced.

2.2.14.4 SESSION HARD-TIMEOUT

Configures the timeout to deauthenticate a session, regardless of the activity carried out by the subscriber. The default value is 1 hour.

Syntax:

```
Network wlan0/0>session hard-timeout <value>
```

Example:

```
Network wlan0/0>session hard-timeout 2h
```

To set the default timeout value to deauthenticate a session, regardless of subscriber activity, enter **no session hard-timeout**.

Command history:

Release	Modification
11.00.03	The "*session hard-timeout*" command was introduced as of version 11.00.03.

2.2.14.5 SESSION IDLE-TIMEOUT

Configures a timeout to deauthenticate a session if no subscriber activity is detected.

Syntax:

```
Network wlan0/0>session idle-timeout <value>
```

Example:

```
Network wlan0/0>session idle-timeout 45m
```

To delete the timeout to deauthenticate a session when no subscriber activity is detected, enter **no session idle-timeout**.

Command history:

Release	Modification
11.00.03	The "*session idle-timeout*" command was introduced as of version 11.00.03.

2.2.14.6 SESSION MAX-OCTETS-DOWN

Configures the maximum number of bytes per-session that can be transmitted in a downlink direction.

Syntax:

```
Network wlan0/0>session max-octets-down <value>
```

Example:

```
Network wlan0/0>session max-octets-down 1048576
```

To delete the maximum number of bytes per-session to transmit in a downlink direction, enter **no session max-octets-down**.

Command history:

Release	Modification
11.00.03	The "*session max-octets-down*" command option was introduced as of version 11.00.03.

Release	Modification
11.00.05, 11.01.01	This command option has become obsolete.

2.2.14.7 SESSION MAX-OCTETS-UP

Configures the maximum number of bytes per-session that can be received in an uplink direction.

Syntax:

```
Network wlan0/0>session max-octets-up <value>
```

Example:

```
Network wlan0/0>session max-octets-up 262144
```

To delete the maximum number of bytes per-session to be received in an uplink direction, enter **no session max-octets-up**.

Command history:

Release	Modification
11.00.03	The "*session max-octets-up*" command option was introduced as of version 11.00.03.
11.00.05, 11.01.01	This command option has become obsolete.

2.2.14.8 SESSION UPLINK

Configures uplink session parameters. The configuration set under this section shall apply to the IP traffic that reaches the router from authenticated subscribers.

Syntax:

```
Network wlan0/0>session uplink ?
  bandwidth    Bandwidth control parameters
  qos          Quality of Service parameters
```

Command history:

Release	Modification
11.00.05, 11.01.01	The "*session uplink*" command option was introduced.

2.2.14.8.1 SESSION UPLINK BANDWIDTH

Shows all the configurable parameters that enable bandwidth limitation per authenticated subscriber. This feature is based on a traffic-shaping algorithm, similar to the one found in the BRS feature. For more detailed information on how it works, please see *Dm715-I BRS Feature.*

Syntax:

```
Network wlan0/0> session uplink bandwidth ?
  burst-committed    Maximum burst committed
  burst-excess       Maximum burst excess
  kbps               Maximum transmission rate
```

Command history:

Release	Modification
11.00.05, 11.01.01	The "*session uplink bandwidth*" command option was introduced.

2.2.14.8.1.1 SESSION UPLINK BANDWIDTH BURST-COMMITTED

Configures the maximum burst size allowed when rate limit is enabled. The value is specified in bytes. If no value is configured, this parameter will be set to an eighth part of the maximum rate. The ratio between the rate limit and the present parameter can be understood as the time interval during which the configured speed is guaranteed on average. For example, if you want to set a certain admissible rate at half-second intervals, the value configured must be half of the configured rate. The time interval must be set between 7.8ms and 1s.

Syntax:

```
Network wlan0/0>session uplink bandwidth burst-committed <value>
```

Example:

```
Network wlan0/0>session uplink bandwidth burst-committed 1000
```

To delete a bandwidth limit per-session for uplink traffic, enter **no session uplink bandwidth burst-committed**.

Command history:

Release	Modification
11.00.05, 11.01.01	The "*session uplink bandwidth burst-committed*" command option was introduced.

2.2.14.8.1.2 SESSION UPLINK BANDWIDTH BURST-EXCESS

Configures the burst excess allowed when rate limit is enabled. The value is specified in bytes. If no value is configured, this parameter will be set to zero. This parameter adds a margin when the rate is reached, making it possible to adjust the desirable rate limit in different scenarios.

Syntax:

```
Network wlan0/0>session uplink bandwidth burst-excess <value>
```

Example:

```
Network wlan0/0>session uplink bandwidth burst-excess 300
```

To delete a bandwidth limit per-session for uplink traffic, enter **no session uplink bandwidth burst-excess**.

Command history:

Release	Modification
11.00.05, 11.01.01	The "*session uplink bandwidth burst-excess*" command option was introduced.

2.2.14.8.1.3 SESSION UPLINK BANDWIDTH KBPS

Configures a bandwidth limit per session for uplink traffic in Kbps. In addition to setting the maximum rate per session, this command enables or disables the rate limit feature.

Syntax:

```
Network wlan0/0>session uplink bandwidth kbps <value>
```

Example:

```
Network wlan0/0>session uplink bandwidth kbps 2000
```

To delete a bandwidth limit per-session for uplink traffic, enter **no session uplink bandwidth kbps**.

Command history:

Release	Modification
11.00.05, 11.01.01	The "*session uplink bandwidth kbps*" command option was introduced.

2.2.14.8.2 SESSION UPLINK QOS

Shows all the configurable parameters that can be used to enable QoS in each authenticated subscriber.

Syntax:

```
Network wlan0/0> session uplink qos ?
  max-octets      Maximum of total octets transmitted
  queue-length    Packet queue length
```

Command history:

Release	Modification
11.00.05, 11.01.01	The "*session uplink qos*" command option was introduced.

2.2.14.8.2.1 SESSION UPLINK QOS MAX-OCTETS

Configures the maximum data volume that an authenticated subscriber can transmit in a session. The value is specified in bytes. If no value is configured, data volume restriction is not applied. Whenever the subscriber reaches the available credit, all incoming IP traffic is dropped by this feature.

Syntax:

```
Network wlan0/0>session uplink qos max-octets <value>
```

Example:

```
Network wlan0/0>session uplink qos max-octets 300000000
```

To delete a bandwidth limit per-session for uplink traffic, enter **no session uplink qos max-octets**.

Command history:

Release	Modification
11.00.05, 11.01.01	The "*session uplink qos max-octets*" command option was introduced.

2.2.14.8.2.2 SESSION UPLINK QOS QUEUE-LENGTH

Configures the packet queue size. This value represents the number of packets the queue can accept. If no value is configured, this parameter will be set to ten.

Syntax:

```
Network wlan0/0>session uplink qos queue-length <value>
```

Example:

```
Network wlan0/0>session uplink qos queue-length 50
```

To delete a bandwidth limit per-session for uplink traffic, enter **no session uplink qos queue-length**.

Command history:

Release	Modification
11.00.05, 11.01.01	The "*session uplink qos queue-length*" command option was introduced.

2.2.15 STATUS-API-REQUEST

Sends an API HTTP/HTTPS request to the client and checks the HotSpot status.

Syntax:

```
Network wlan0/0>status-api-request ?
  url             Configure url to send HTTP/HTTPS status API-request from hotspot
  source-address  Configure a source-address for API-request
  parameter       Configure parameter of API-request
```

2.2.15.1 STATUS-API-REQUEST URL

A URL is configured in order to send an API HTTP/HTTPS request to the client and check the status of the HotSpot.

Syntax:

```
Network wlan0/0>status-api-request url <url>
```

Example:

```
Network wlan0/0>status-api-request url https://www.test.com
```

To delete a configured URL, enter **no status-api-request url <url>**.

Command history:

Release	Modification
11.01.07	The "*status-api-request url*" command was introduced as of version 11.01.07.
11.01.08	The "*status-api-request url*" command was modified as of version 11.01.08

2.2.15.2 STATUS-API-REQUEST PARAMETER

Configuration of the necessary parameters for an API request. For instance, a key.

Syntax:

```
Network wlan0/0>status-api-request parameter <text> value <text>
```

Example:

```
Network wlan0/0>status-api-request parameter key value 123456
```

To delete a configured parameter, enter **no status-api-request parameter <text> value <text>**.

Command history:

Release	Modification
11.01.07	The "*status-api-request parameter*" command was introduced as of version 11.01.07.
11.01.08	The "*status-api-request parameter*" command was modified as of version 11.01.08

2.2.15.3 STATUS-API-REQUEST SOURCE-ADDRESS

Source address to route API/Hotspot traffic.

Syntax:

```
Network wlan0/0>status-api-request source-address <ip>
```

Example:

```
Network wlan0/0>status-api-request source-address 192.168.1.1
```

To delete a configured source-address, enter **no status-api-request source-address <ip>**.

Command history:

Release	Modification
11.01.08	The "*status-api-request source-address*" command was introduced as of version 11.01.08.

2.2.16 UAM-SERVER

Configures the UAM server feature.

Syntax:

```
Network wlan0/0>uam-server ?
  address           Configure IP address for UAM server
  authentication    Configure options for UAM authentication
  port              Configure TCP port for UAM server
  secure            Configure UAM secure server (HTTPS)
```

Command history:

Release	Modification
11.00.03	The "*uam-server*" command was introduced as of version 11.00.03.

2.2.16.1 UAM-SERVER ADDRESS

Configures the IP address used by the UAM server.

Syntax:

```
Network wlan0/0>uam-server address <IPv4 address>
```

Example:

```
Network wlan0/0>uam-server address 172.16.0.254
```

To delete the IP address used by the UAM server, enter **no uam-server address**.

Command history:

Release	Modification
11.00.03	The "*uam-server address*" command was introduced as of version 11.00.03.

2.2.16.2 UAM-SERVER AUTHENTICATION

Configures the authentication options in the UAM server.

Syntax:

```
Network wlan0/0>uam-server authentication {chap | domain <Domain text> | secret <Secret text>}
```

Command history:

Release	Modification
11.00.03	The "*uam-server authentication*" command was introduced as of version 11.00.03.

2.2.16.2.1 UAM-SERVER AUTHENTICATION CHAP

Enables CHAP authentication. A challenge is generated for every subscriber found. This challenge is valid for 10 minutes.

Syntax:

```
Network wlan0/0>uam-server authentication chap
```

To disable CHAP authentication, enter **no uam-server authentication chap**.

Command history:

Release	Modification
11.00.03	The "*uam-server authentication chap*" command was introduced as of version 11.00.03.

2.2.16.2.2 UAM-SERVER AUTHENTICATION DOMAIN

Configures a domain to be added to the username in authentication requests sent to the AAA server.

Syntax:

```
Network wlan0/0>uam-server authentication domain <Domain text>
```

Example:

```
Network wlan0/0>uam-server authentication domain @fec_rd.de
```

To delete a domain to be added to the username in authentication requests, enter **no uam-server authentication domain**.

Command history:

Release	Modification
11.00.03	The "*uam-server authentication domain*" command was introduced as of version 11.00.03.

2.2.16.2.3 UAM-SERVER AUTHENTICATION SECRET

Configures a shared secret between the UAM server and the external captive portal server.

Syntax:

```
Network wlan0/0>uam-server authentication secret <Secret text>
```

Example:

```
Network wlan0/0>uam-server authentication secret my-secret
```

To delete a shared secret between the UAM server and the external captive portal server, enter **no uam-server authentication secret**.

Command history:

Release	Modification
11.00.03	The "*uam-server authentication secret*" command was introduced as of version 11.00.03.

2.2.16.3 UAM-SERVER PORT

Configures the TCP port used by the UAM server. The default value is 4532.

Syntax:

```
Network wlan0/0>uam-server port <TCP port>
```

Example:

```
Network wlan0/0>uam-server port 80
```

To set the default value for the TCP port used by the UAM server, enter **no uam-server port**.

Command history:

Release	Modification
11.00.03	The "*uam-server port*" command was introduced as of version 11.00.03.

2.2.16.4 UAM-SERVER SECURE

Configures the HTTPS feature options for the UAM server.

Syntax:

```
Network wlan0/0>uam-server secure {ca <Certificate name> | dont-verify | enable | port <TCP port> | user <Certificate name>}
```

Command history:

Release	Modification
11.00.03	The "*uam-server secure*" command was introduced as of version 11.00.03.

2.2.16.4.1 UAM-SERVER SECURE CA

Adds a certificate issued by a Certification Authority that was previously loaded in the router. Please see manual *Dm739-I IPSec* for further information on certificate loading.

Syntax:

```
Network wlan0/0>uam-server secure ca <Certificate name>
```

Example:

```
Network wlan0/0>uam-server secure ca SERVER_A.CER
```

To delete a Certification Authority certificate, enter **no uam-server secure ca <Certificatename>**.

Command history:

Release	Modification
11.00.03	The "*uam-server secure ca*" command was introduced as of version 11.00.03.

2.2.16.4.2 UAM-SERVER SECURE DONT-VERIFY

Prevents the device from requesting an X509 certificate from connected clients so that said clients are not authenticated. This command is disabled by default, meaning the device does request certification from clients and checks they are signed by a secure certificate authority.

Syntax:

```
Network wlan0/0>uam-server secure dont-verify
```

To request a certificate from clients, enter **no uam-server secure dont-verify**.

Command history:

Release	Modification
11.00.07, 11.01.02	The "*uam-server secure dont-verify*" command was introduced.

2.2.16.4.3 UAM-SERVER SECURE ENABLE

Enables HTTPS in the UAM server.

Syntax:

```
Network wlan0/0>uam-server secure enable
```

To disable HTTPS in the UAM server, enter **no uam-server secure enable**.

Command history:

Release	Modification
11.00.03	The "*uam-server secure enable*" command was introduced as of version 11.00.03.

2.2.16.4.4 UAM-SERVER SECURE PORT

Configures the TCP port used by HTTPS. The default value is 443.

Syntax:

```
Network wlan0/0>uam-server secure port <TCP port>
```

Example:

```
Network wlan0/0>uam-server secure port 443
```

To set a default value for the TCP port used by HTTPS, enter **no uam-server secure port**.

Command history:

Release	Modification
11.00.03	The "*uam-server secure port*" command was introduced as of version 11.00.03.

2.2.16.4.5 UAM-SERVER SECURE USER

Configures the user certificate sent by the server in an HTTPS communication. This certificate must be previously loaded in the router. Please see manual *Dm739-I IPSec* for further information on certificate loading.

Syntax:

```
Network wlan0/0>uam-server secure user <Certificate name>
```

Example:

```
Network wlan0/0>uam-server secure user SERVER.CER
```

To delete the user certificate the server sends in a HTTPS communication, enter **no uam-server secure user**.

Command history:

Release	Modification
11.00.03	The "*uam-server secure user*" command was introduced as of version 11.00.03.

2.2.17 URL

Configures the URLs to which the client is redirected to by UAM server responses.

Syntax:

```
Network wlan0/0>url {fail-page <URL> | portal-page <URL> | query-format <Option> | success-page <URL>}
```

Command history:

Release	Modification
11.00.03	New command added.
11.00.04, 11.01.00	New **query-format** command added.

Release	Modification
11.00.05, 11.01.00, 11.00.04.02.02	New **query-format field-nas-id** command added.
11.00.06, 11.01.01, 11.00.04.02.02	New **nas-id** and **user-mac** command options added under success-page.

2.2.17.1 URL FAIL-PAGE

Configures the URL to which a client is redirected to after a failed authentication process.

Syntax:

```
Network wlan0/0>url fail-page <URL text>
```

Example:

```
Network wlan0/0>url fail-page http://172.18.0.1/fail.html
```

To delete the URL to which a client is being redirected to after failed authentication, enter **no url fail-page**.

Command history:

Release	Modification
11.00.03	The "*url fail-page*" command was introduced as of version 11.00.03.

2.2.17.2 URL PORTAL-PAGE

Configures the captive portal URL to which a client is being redirected to initialize authentication.

Syntax:

```
Network wlan0/0>url portal-page <URL text>
```

Example:

```
Network wlan0/0>url portal-page http://172.18.0.1/login.html
```

To delete the captive portal URL to which a client is being redirected to initialize authentication, enter **no urlportal-page**.

Command history:

Release	Modification
11.00.03	The "*url portal-page*" command was introduced as of version 11.00.03.

2.2.17.3 URL QUERY-FORMAT

Configures the optional parameters included in the URL used for initial HTTP redirection of *unauthorized* subscribers. This URL is built using the configured portal page, extending it with a set of field-value pairs in a URL query format.

Syntax:

```
Network wlan0/0>url query-format ?
  coa-ip          Include the CoA address field
  coa-port        Include the CoA port field
  field-nas-id    Set NAS-ID field
  loginurl        Loginurl as the unique parameter in URL query
```

Command history:

Release	Modification
11.00.04, 11.01.00	The "*url query-format*" command was introduced.
11.00.05, 11.01.00, 11.00.04.02.02	The " **field-nas-id**" command option was introduced.
11.00.06, 11.01.02	The "*coa-ip*" and "*coa-port*" command options were introduced.

2.2.17.3.1 URL QUERY-FORMAT COA-IP

Adds the *coaip* parameter to the URL built to redirect clients to the captive portal. This parameter must be the IP address used by the router to listen for CoA requests.

Syntax:

```
Network wlan0/0>url query-format coa-ip <IP address>
```

Example:

```
Network wlan0/0>url query-format coa-ip 10.0.0.1
```

To stop the *coaip* parameter from being included in the URL, enter **no url query-format coa-ip**.

Command history:

Release	Modification
11.00.06, 11.01.02	The "*url query-format coa-ip*" command was introduced.

2.2.17.3.2 URL QUERY-FORMAT COA-PORT

Adds the *coaport* parameter to the URL built to redirect clients to the captive portal. This parameter must be the UDP port used by the router to listen for CoA requests.

Syntax:

```
Network wlan0/0>url query-format coa-port <Port number>
```

Example:

```
Network wlan0/0>url query-format coa-port 3799
```

To stop the *coaport* parameter from being included in the URL, enter **no url query-format coa-port**.

Command history:

Release	Modification
11.00.06, 11.01.02	The "*url query-format coa-port*" command was introduced.

2.2.17.3.3 URL QUERY-FORMAT LOGINURL

Configures the **loginurl** parameter that must be included in the URL used for initial HTTP redirection. The value of this parameter is a URL encoded string, with the field-value pairs included in the default redirection.

This URL format may be used when initial redirection is done towards a splash page, which redirects clients to a secondary server.

Syntax:

```
Network wlan0/0>url query-format loginurl
```

If you do not wish to use the **loginurl** parameter, enter **no url query-format loginurl**.

Command history:

Release	Modification
11.00.04, 11.01.00	The "*url query-format loginurl*" command was introduced.

2.2.17.3.4 URL QUERY-FORMAT FIELD-NAS-ID

Inserts the **nas-id** field in the URL query. This field is set with a string that identifies the device with a short name. This way, the hotspot gateway shares its identity with the captive portal server.

Syntax:

```
Network wlan0/0>url query-format field-nas-id <value>
```

Example:

```
Network wlan0/0>url query-format field-nas-id hs_gateway_03
```

If you do not wish to add this field, enter **no url query-format field-nas-id**.

Command history:

Release	Modification
11.00.05, 11.01.00, 11.00.04.02.02	The "*url query-format field-nas-id*" command was introduced.

2.2.17.4 URL SUCCESS-PAGE

Configures the URL to which a client is redirected to after successful authentication.

Syntax:

```
Network wlan0/0>url success-page <URL text>
```

Example:

```
Network wlan0/0>url success-page http://172.18.0.1/success.html
```

To delete the URL to which a client is redirected to after successful authentication, enter **no url success-page**.

Command history:

Release	Modification
11.00.03	The "*url success-page*" command was introduced as of version 11.00.03.
11.00.06, 11.01.01, 11.00.04.02.02	The "*nas-id*" and "*user-mac*" command options were introduced.

2.2.17.4.1 URL SUCCESS-PAGE NAS-ID

Adds a field-value pair to the URL query in the successful page, which identifies the router responsible for the redirection. If the query does not exist, the command automatically adds a question mark after the string to create it. Since the URL may already have a query, this field is concatenated with the '&' character after the last pair.

Syntax:

```
Network wlan0/0>url success-page nas-id value <NAS-ID value> field <NAS-ID field name>
```

Example:

```
Network wlan0/0>url success-page nas-id value hotspot_nas_03 field NAS-ID
```

The command in the last example will add the 'NAS-ID=hotspot_nas_03' pair to the successful URL query. If you do not wish to add this field, enter **no url success-page nas-id**.

Command history:

Release	Modification
11.00.06, 11.01.01, 11.00.04.02.02	The "*url success-page nas-id*" command was introduced.

2.2.17.4.2 URL SUCCESS-PAGE USER-MAC

Adds a field-value pair to the URL query in the successful page, which identifies the user that has been authenticated by the device MAC address. If the query does not exist, the command automatically adds a question mark after the string to create it. Since the URL may already have a query, this field is concatenated with the '&' character after the last pair.

Syntax:

```
Network wlan0/0>url success-page user-mac spacer <spacer char> field <user-mac field name>
```

Example:

```
Network wlan0/0>url success-page user-mac spacer - field userMac
```

The command in the last example will add the 'userMac=xx-xx-xx-xx-xx-xx' pair to the successful URL query. If you do not wish to add this field, enter **no url success-page user-mac**.

Command history:

Release	Modification
11.00.06, 11.01.01, 11.00.04.02.02	The "*url success-page user-mac*" command was introduced.

2.2.18 WALLED-GARDEN

Configures IP traffic criteria, even if it comes from unauthenticated subscribers. The traffic that matches some of the walled garden criteria is not redirected to the UAM server and no HotSpot access policies are applied.

Syntax:

```
Network wlan0/0>walled-garden {access-list <ACL number> | domain <name>}
```

Command history:

Release	Modification
11.00.03	The "*walled-garden*" command was introduced as of version 11.00.03.
11.00.05, 11.01.00	New **domain** command added.

2.2.18.1 WALLED-GARDEN ACCESS-LIST

Turns an existing access list into a walled garden list for the current HotSpot session. The list may be standard or extended, but its type must always allow for configurations that include walled garden domain names. In such cases, the list not only contains the configured entries, but also dynamic entries added by domain names. We recommend configuring several lists, in case there are multiple hotspots running at the same time.

Syntax:

```
Network wlan0/0>walled-garden access-list <ACL number>
```

Example:

```
Network wlan0/0>walled-garden access-list 101
```

To delete the access list used to define the walled garden, enter **no walled-garden access-list**.

Command history:

Release	Modification
11.00.03	The "*walled-garden access-list*" command was introduced as of version 11.00.03.

2.2.18.2 WALLED-GARDEN DOMAIN

Allows for a list of domain names to be configured and included in the walled garden. IP traffic with a destination domain that matches a list entry can pass through the router, even if it comes from unauthenticated users. Adding an asterisk (*) at the front of a domain allows for all sub-domains to pass as well. To view what destination IP addresses users may reach, display all list entries from the access list monitoring menu. Every entry has a description field that links the canonical domain to the IP address. See manual *Dm752-I Access Control* for further information.

Note

This feature only works properly if an extended access list is configured as walled garden and the router is set up as a DNS server for HotSpot users.

Syntax:

```
Network wlan0/0>walled-garden domain <name>
```

Example:

```
Network wlan0/0>walled-garden domain a.domain.com
Network wlan0/0>walled-garden domain *domain.com
```

To delete a domain, just enter **no walled-garden domain <name>**.

Command history:

Release	Modification
11.00.05. 11.01.00	The "*walled-garden domain*" command was introduced.

2.2.19 WHITE-LIST

Adds a subscriber MAC address to the white-list. Traffic sent by said subscriber is not redirected to the UAM server and no HotSpot access policies are applied.

Syntax:

```
Network wlan0/0>white-list mac <MAC address>
```

Example:

```
Network wlan0/0>white-list mac 00-11-22-33-44-55
```

To delete a MAC address from the white-list, enter **no white-list mac <MAC address>**.

Command history:

Release	Modification
11.00.03	The "*white-list*" command was introduced as of version 11.00.03.

2.2.20 EXIT

Exits the network configuration menu.

Command history:

Release	Modification
11.00.03	The "*exit*" command was introduced as of version 11.00.03.

Chapter 3 Monitoring

3.1 Global Monitoring

To access the monitoring menu for a HotSpot, use the **feature hotspot** command (found on the main monitoring menu).

Example:

```
+feature hotspot

--  Hotspot User Console  --
HS+
```

In the HotSpot feature's global monitoring menu, the following commands are available:

Command	Function
? (HELP)	Displays the monitoring commands and their options.
NETWORK	Monitors an interface HotSpot.
EXIT	Exits the HotSpot monitoring menu.

3.1.1 ? (HELP)

Displays the available commands and their options.

Command history:

Release	Modification
11.00.03	The "*? (Help)*" command was introduced as of version 11.00.03.

3.1.2 NETWORK

Accesses the HotSpot monitoring menu for a specific interface.

Syntax:

```
HS+network <Interface name>
```

Example:

```
HS+network wlan0/0
HS [wlan0/0]+
```

Command history:

Release	Modification
11.00.03	The "*network*" command was introduced as of version 11.00.03.

3.1.3 EXIT

Exits the monitoring menu of the HotSpot feature.

Command history:

Release	Modification
11.00.03	The "*exit*" command was introduced as of version 11.00.03.

3.2 Network Monitoring

Monitors the HotSpot feature for a specific interface. The following table shows the commands available:

Command	Function
? (HELP)	Displays the monitoring commands and their options.

LIST	Lists the HotSpot client statistics in the interface.
STATISTICS	Displays the HotSpot statistics in the interface.
EXIT	Exits the WNMS monitoring menu.

Although HotSpot can be configured in several networks, `wlan0/0` will be used from here onwards in the examples and syntax rules included in the following sections.

3.2.1 ? (HELP)

Displays the commands available and their options.

Command history:

Release	Modification
11.00.03	The "*? (Help)*" command was introduced as of version 11.00.03.

3.2.2 LIST

Displays the HotSpot client statistics found for the interface.

Syntax:

```
HS [wlan0/0]+list clients [ip <IPv4 address>] [status {authenticated | unauthenticated}] /
                          [mac <aa-bb-cc-dd-ee-ff>] [low-detail | medium-detail | high-detail]
```

Filtering options are as follows:

- `ip`: option to filter through the IP addresses of subscribers.
- `status`: option to filter through the session status of subscribers.
- `mac`: option to filter through the MAC address of subscribers.
- `low-detail`, `medium-detail`, `high-detail`: option to set the level of detail of the information. By default, `medium-detail` is always used.

Example 1:

```
HS [wlan0/0]+list clients

============================
..::  Hotspot clients  ::..
============================

Client IP: 172.16.0.9, Subscriber MAC: 00-16-36-03-90-8d, Discovered: 11/19/2014 16:53:22
Status: Unauthenticated

Client IP: 172.16.0.10, Subscriber MAC: 68-9c-5e-b7-9b-4d, Discovered: 11/19/2014 17:03:56
Status: Authenticated, Session ID: 14BA45B80005, Session User: tel1
Session init: 11/19/2014 17:04:21, Remaining: 59m57s
Session downlink bandwidth Kbps: 2000, Bc: 2000, Be: 0
Session uplink bandwidth Kbps: 1000, Bc: 1000, Be: 0
Session downlink QoS max octets: 1000000000, queue length: 10
Session uplink QoS max octets: 500000000, queue length: 10
Output octets: 321624, Input octets: 35467
Output octets by IP: 321624, Input octets by IP: 35467
Drop bytes by BW limit downlink: 0, Drop bytes by BW limit uplink: 0
Drop bytes by max credit downlink: 0, Drop bytes by max credit uplink: 0

HS [wlan0/0]+
```

Example 2:

```
HS [wlan0/0]+list clients ip 172.16.0.10

============================
..::  Hotspot clients  ::..
============================

Client IP: 172.16.0.10, Subscriber MAC: 68-9c-5e-b7-9b-4d, Discovered: 11/19/2014 17:03:57
```

```
Status: Authenticated, Session ID: 14BA45B80005, Session User: tel1
Session init: 11/19/2014 17:04:21, Remaining: 59m56s
Session downlink bandwidth Kbps: 2000, Bc: 2000, Be: 0
Session uplink bandwidth Kbps: 1000, Bc: 1000, Be: 0
Session downlink QoS max octets: 1000000000, queue length: 10
Session uplink QoS max octets: 500000000, queue length: 10
Output octets: 321624, Input octets: 35467
Output octets by IP: 321624, Input octets by IP: 35467
Drop bytes by BW limit downlink: 0, Drop bytes by BW limit uplink: 0
Drop bytes by max credit downlink: 0, Drop bytes by max credit uplink: 0

HS [wlan0/0]+
```

Command history:

Release	Modification
11.00.03	The "*list*" command was introduced as of version 11.00.03.
11.00.07, 11.01.02	The "*mac*", "*low-detail*", "*medium-detail*" and "*high-detail*" options have been added.

3.2.3 STATISTICS

Displays HotSpot statistics for the interface.

Syntax:

```
HS+statistics
```

Example:

```
HS [wlan0/0]+statistics

============================
..::  Hotspot statistics ::..
============================

Enabled uptime: 2d0h13m31s
Discovered subscribers: 12
Authenticated subscribers: 2
Discovered IP clients: 12
Active HTTP UAM sessions: 1

HS [wlan0/0]+
```

Command history:

Release	Modification
11.00.03	The "*statistics*" command was introduced as of version 11.00.03.

3.2.4 EXIT

Exits network monitoring.

Command history:

Release	Modification
11.00.03	The "*exit*" command was introduced as of version 11.00.03.

Chapter 4 Configuration Examples

This section provides configuration examples for different types of scenario.

The following configuration steps are recommended:

(a) AAA configuration: configures server parameters for AAA services.

(b) Walled garden configuration: defines IP criteria for the walled garden.

(c) HotSpot configuration: configures the UAM server function, session parameters, AAA services and access policies.

(d) Filtering configuration (optional): depending on the HotSpot access policy, *non-accepted* packets can be discarded through the access groups.

4.1 Example 1: captive portal in the local network

The scenario for this example presents both AAA and captive portal servers in a local area network.

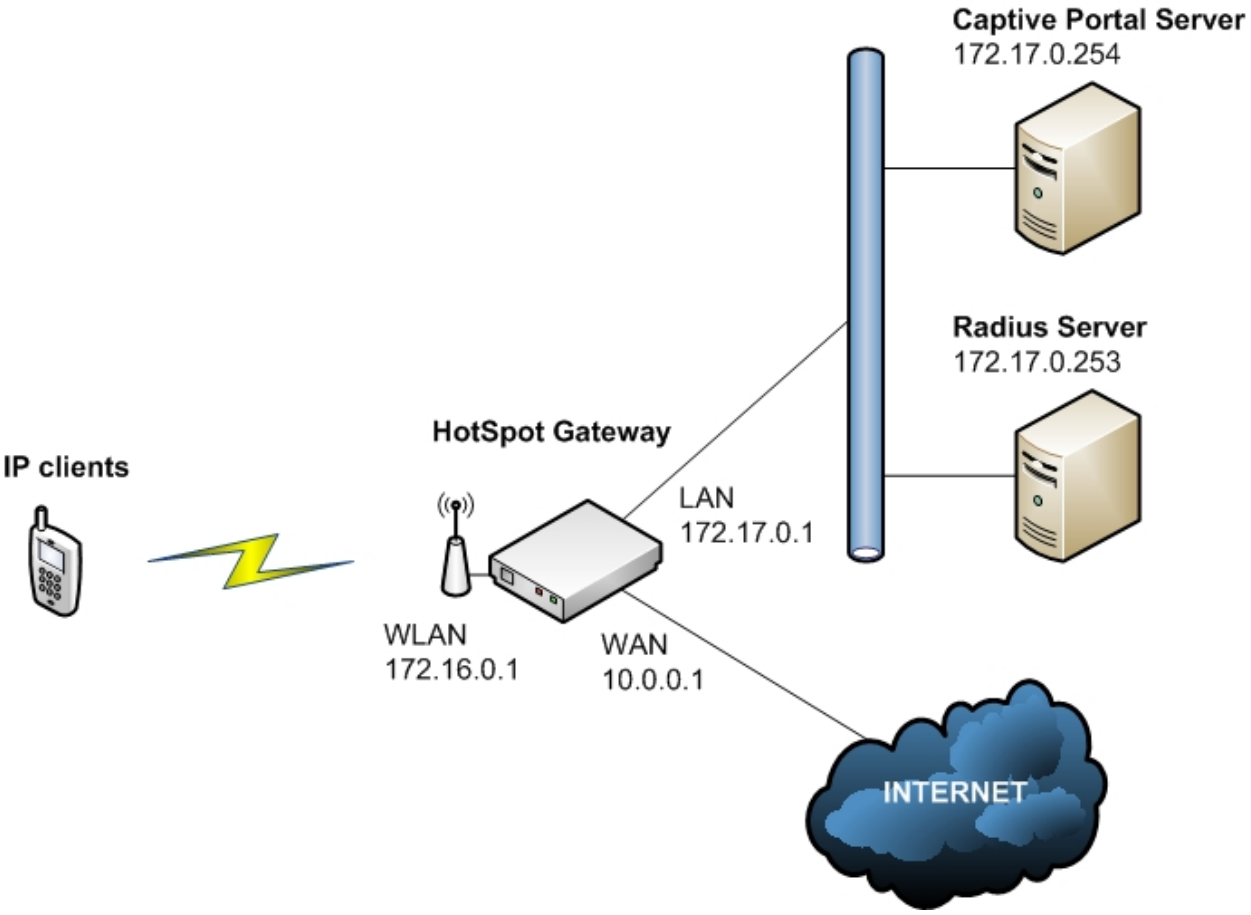

Fig. 4: **Example 1: Network diagram**

Wireless devices attempt to access the Internet through the WLAN interface (wlan0/0). To reach Internet destinations, traffic is routed to the WAN interface (ethernet0/1). To obtain access, devices must be authenticated in the HotSpot function. The LAN interface (ethernet0/0) is used to provide access to servers used by the HotSpot.

4.1.1 Radius configuration

A Radius server, running in 172.17.0.253, is configured using the Radius feature for authentication. Radius *access-accept* response can include specific session parameters per user. Please see manual *Dm733-I Radius Protocol* for further information.

Configuration example:

```
   feature radius
; -- RADIUS User Configuration --
      primary-address 172.17.0.253
```

```
      primary-secret my-radius-secret
      enable radius
   exit
```

4.1.2 AFS configuration

An AFS feature is enabled in this scenario. It allows the matching of *non-accepted* packets in stateful access lists through the use of **entry <id> subscriber-status unauthenticated** . Please see manual *Dm786-I AFS* for further information on the Advanced Firewall System.

Configuration example:

```
   feature afs
      enable
   exit
```

4.1.3 IPSec configuration

To enable HTTPS in the UAM server, all necessary certificates must be preloaded in the IPSec configuration menu. Please see manual *Dm739-I IPSec* for further information on loading certificates.

Example configuration:

```
      ipsec
; -- IPSec user configuration --
         cert
; -- Cert user configuration --
            file b64new SERVER.CER
            file add "-----BEGIN CERTIFICATE-----"
            file add MIIC+zCCAmSgAwIBAgIBBDANBgkqhkiG9w0BAQUFADCBhDELMAkGA1UEBhMCRVMx
            file add DzANBgNVBAgMBk1hZHJpZDEPMA0GA1UEBwwGTWFkcmlkMQ8wDQYDVQQKDAZUZWxk
            file add YXQxDzANBgNVBAsMBlRlbGRhdDEPMA0GA1UEAwwGVGVsZGF0MSAwHgYJKoZIhvcN
            file add AQkBFhF0ZWxkYXRAdGVsZGF0LmNvbTAeFw0xNDEwMDkxMzU3MjZaFw0xNTEwMDkx
            file add MzU3MjZaMIGEMQswCQYDVQQGEwJFUzEPMA0GA1UECAwGTWFkcmlkMQ8wDQYDVQQH
            file add DAZNYWRyaWQxDzANBgNVBAoMBlRlbGRhdDEPMA0GA1UECwwGVGVsZGF0MQ8wDQYD
            file add VQQDDAZUZWxkYXQxIDAeBgkqhkiG9w0BCQEWEXRlbGRhdEB0ZWxkYXQuY29tMIGf
            file add MA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDT2gwiddeALnEFgf82LFAWtqxqpw/p
            file add iDV4bB40ZW5792BwKRuS8L9doLFWcQG0UyxbgyXVWI7+B/0+rRi3jyC7aOM8/we6
            file add KOdOnrTisLz8FTOylz8SPYFatdhaESGAqi4WZmjgQ/DIdwOrmEECW003l3DIj1KZ
            file add MOJ9IO/EH9cr6QIDAQABo3sweTAJBgNVHRMEAjAAMCwGCWCGSAGG+EIBDQQfFh1P
            file add cGVuU1NMIEdlbmVyYXRlZCBDZXJ0aWZpY2F0ZTAdBgNVHQ4EFgQUd1sDRbGs6Vuu
            file add TsahuBGQ0Kjea9UwHwYDVR0jBBgwFoAUN75qYWs23kPJiuSpV2tq59RMvKwwDQYJ
            file add KoZIhvcNAQEFBQADgYEAmQbxnOekpSe2ikOSJCw3Sp/9zFC3qDYBzlIatcba6FQ0
            file add hrC5knl2EiuZK55FTe+kD4RbfMks1AHaqn3oKscUQ67FKRQaC8bdWr3ZoU1U9QY6
            file add dPj6JvFQml+cnclmhb5Qbui/WFU8F0hWB4kgpcs7VqGEkjAj0uHDNLkX8D41kkM=
            file end "-----END CERTIFICATE-----"
            file b64new SERVER_A.CER
            file add "-----BEGIN CERTIFICATE-----"
            file add MIIC2DCCAkGgAwIBAgIJANzplcbFEjVIMA0GCSqGSIb3DQEBBQUAMIGEMQswCQYD
            file add VQQGEwJFUzEPMA0GA1UECAwGTWFkcmlkMQ8wDQYDVQQHDAZNYWRyaWQxDzANBgNV
            file add BAoMBlRlbGRhdDEPMA0GA1UECwwGVGVsZGF0MQ8wDQYDVQQDDAZUZWxkYXQxIDAe
            file add BgkqhkiG9w0BCQEWEXRlbGRhdEB0ZWxkYXQuY29tMB4XDTE0MTAwODEwMDk1M1oX
            file add DTI0MTAwNTEwMDk1M1owgYQxCzAJBgNVBAYTAkVTMQ8wDQYDVQQIDAZNYWRyaWQx
            file add DzANBgNVBAcMBk1hZHJpZDEPMA0GA1UECgwGVGVsZGF0MQ8wDQYDVQQLDAZUZWxk
            file add YXQxDzANBgNVBAMMBlRlbGRhdDEgMB4GCSqGSIb3DQEJARYRdGVsZGF0QHRlbGRh
            file add dC5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBALoenHsvxFexXtKKtN0q
            file add fpYNo/t/epcsXIXIssKH6Eqq3YwMw1zjZfLhmi5W5MYqFUB/vW2fJHwArgb6oJk+
            file add 91230ipPrXSV5InMIeqw6JQMJb01RCWQCVVMu+2uAAeMP700dSIOwUnHfi/P3KmW
            file add pNS7adPQh10DdVumcv7Sn63zAgMBAAGjUDBOMB0GA1UdDgQWBBQ3vmphazbeQ8mK
            file add 5KlXa2rn1Ey8rDAfBgNVHSMEGDAWgBQ3vmphazbeQ8mK5KlXa2rn1Ey8rDAMBgNV
            file add HRMEBTADAQH/MA0GCSqGSIb3DQEBBQUAA4GBAKsRJRvTNVdJIxAvVdhPyF2GJZeZ
            file add hb3dBDZ11Bt145A3zof0zxaxBo4B/PnrHNeJ60I79uFBmNRQxAi7pa6J7ow+a0jq
            file add dOfPzmnv6Yz980OdWRmRbfbjC1mul3ZUTdXD+Sq8TX7cG46xqt23wuWHx1ql6tkg
            file add O5aHMHKpCrzyWCnq
            file end "-----END CERTIFICATE-----"
;
            certificate SERVER_A.CER load
```

```
            certificate SERVER.CER load
         exit
;
      exit
;
```

4.1.4 Walled garden configuration

An access list must be configured to include *permit* entries for the captive portal server destination. The remaining packets must be classified as *non-accepted,* adding *deny* as a last entry so they are subsequently filtered.

Configuration example:

```
   feature access-lists
; -- Access Lists user configuration --
      access-list 101
         description "Walled Garden"
;
         entry 1 description "Captive Portal Server"
         entry 1 default
         entry 1 permit
         entry 1 destination address 172.17.0.254 255.255.255.255
;
         entry 2 default
         entry 2 deny
;
      exit
;
   exit
```

4.1.5 HotSpot configuration

HotSpot needs to redirect clients to the captive portal server running in 172.17.0.254. By configuring **redirect enable,** the *non-accepted* HTTP traffic is received by the UAM server, which processes the request and returns an HTTP response. The UAM server response redirects the HTTP request to the URL configured through the **url portal-page <URL text>** command.

In this example, user credentials are presented, encrypted, to the UAM server using HTTPS. HTTPS is enabled through **uam-server secure enable.** All necessary certifications must be preloaded in the router. To enable a certificate in SSL communication sending, enter the **uam-secure user <Certificate name>** command. Finally, to validate client certificates, a digital certificate issued by a Certification Authority must be added through **uam-server secure ca <Certificate name>.**

Configuration example:

```
   feature hotspot
; -- Hotspot Configuration --
      network wlan0/0
         redirect enable
         uam-server secure enable
         uam-server secure ca SERVER_A.CER
         uam-server secure user SERVER.CER
         url portal-page http://172.17.0.254/login_en.htm
         walled-garden access-list 101
         enable
      exit
;
   exit
```

4.1.6 Filtering configuration

An access group is configured in the WAN interface for outgoing traffic. To achieve this, a stateful access list is used to define the traffic filter and discard the *non-accepted* traffic. In this scenario, the DNS packets, at the very least, must be accepted. Through the **entry <id> subscriber-status unauthenticated** command in a *deny* entry, *non-accepted* packets sent by unauthenticated subscribers are discarded. Please see manual *Dm752-I Access Control* for further information.

Configuration example:

```
   feature access-lists
; -- Access Lists user configuration --
      access-list 5001
         entry 1 default
         entry 1 permit
         entry 1 description "DNS Traffic"
         entry 1 protocol dns
;
         entry 2 default
         entry 2 deny
         entry 2 description "Non-accepted Traffic"
         entry 2 subscriber-status unauthenticated
;
         entry 3 default
         entry 3 permit
;
      exit
;
   exit
```

Through the **ip access.group <id> out** command, the traffic filter is configured in the WAN interface.

Configuration example:

```
   network ethernet0/1
; -- Ethernet Interface User Configuration --
      ip access-group 5001 out
;
;
      ip address 10.0.0.1 255.0.0.0
;
;
   exit
```

4.1.7 Complete configuration

A basic configuration for the proposed example is given below:

```
   log-command-errors
   no configuration
   feature afs
      enable
   exit
;
   feature access-lists
; -- Access Lists user configuration --
      access-list 101
         description "Walled Garden"
;
         entry 1 description "Captive Portal Server"
         entry 1 default
         entry 1 permit
         entry 1 destination address 172.17.0.254 255.255.255.255
;
         entry 2 default
         entry 2 deny
;
      exit
;
      access-list 5001
         entry 1 default
         entry 1 permit
         entry 1 description "DNS Traffic"
         entry 1 protocol dns
;
         entry 2 default
```

```
         entry 2 deny
         entry 2 description "Non-accepted Traffic"
         entry 2 subscriber-status unauthenticated
;
         entry 3 default
         entry 3 permit
;
      exit
;
   exit
;
   network ethernet0/0
; -- Ethernet Interface User Configuration --
;
      ip address 172.17.0.1 255.255.0.0
;
;
   exit
;
;
   network ethernet0/1
; -- Ethernet Interface User Configuration --
      ip access-group 5001 out
;
;
      ip address 10.0.0.1 255.255.0.0
;
;
   exit
;
;
   network wlan0/0
; -- Wireless LAN Interface. Configuration --
      ip address 172.16.0.1 255.255.0.0
;
      bss "my-bss"
         privacy-invoked
         rsn wpa
         cipher aes-ccmp
         akm-suite psk
         wpa-psk passphrase ciphered 0xFCEFFD77D44110F0A8AA0C9B91146913
      exit
;
   exit
;
;
;
;
;
   protocol ip
; -- Internet protocol user configuration --
      route 0.0.0.0 0.0.0.0 10.0.0.254
;
      nat
         rule 1 out ethernet0/1 dynamic overload
         rule 1 translation source interface ethernet0/1
;
      exit
;
      ipsec
; -- IPSec user configuration --
         cert
; -- Cert user configuration --
            file b64new SERVER.CER
            file add "-----BEGIN CERTIFICATE-----"
            file add MIIC+zCCAmSgAwIBAgIBBDANBgkqhkiG9w0BAQUFADCBhDELMAkGA1UEBhMCRVMx
            file add DzANBgNVBAgMBk1hZHJpZDEPMA0GA1UEBwwGTWFkcmlkMQ8wDQYDVQQKDAZUZWxk
```

```
            file add YXQxDzANBgNVBAsMBlRlbGRhdDEPMA0GA1UEAwwGVGVsZGF0MSAwHgYJKoZIhvcN
            file add AQkBFhF0ZWxkYXRAdGVsZGF0LmNvbTAeFw0xNDEwMDkxMzU3MjZaFw0xNTEwMDkx
            file add MzU3MjZaMIGEMQswCQYDVQQGEwJFUzEPMA0GA1UECAwGTWFkcmlkMQ8wDQYDVQQH
            file add DAZNYWRyaWQxDzANBgNVBAoMBlRlbGRhdDEPMA0GA1UECwwGVGVsZGF0MQ8wDQYD
            file add VQQDDAZUZWxkYXQxIDAeBgkqhkiG9w0BCQEWEXRlbGRhdEB0ZWxkYXQuY29tMIGf
            file add MA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDT2gwiddeALnEFgf82LFAWtqxqpw/p
            file add iDV4bB40ZW5792BwKRuS8L9doLFWcQG0UyxbgyXVWI7+B/0+rRi3jyC7aOM8/we6
            file add KOdOnrTisLz8FTOylz8SPYFatdhaESGAqi4WZmjgQ/DIdwOrmEECW003l3DIj1KZ
            file add MOJ9IO/EH9cr6QIDAQABo3sweTAJBgNVHRMEAjAAMCwGCWCGSAGG+EIBDQQfFh1P
            file add cGVuU1NMIEdlbmVyYXRlZCBDZXJ0aWZpY2F0ZTAdBgNVHQ4EFgQUd1sDRbGs6Vuu
            file add TsahuBGQ0Kjea9UwHwYDVR0jBBgwFoAUN75qYWs23kPJiuSpV2tq59RMvKwwDQYJ
            file add KoZIhvcNAQEFBQADgYEAmQbxnOekpSe2ikOSJCw3Sp/9zFC3qDYBzlIatcba6FQ0
            file add hrC5knl2EiuZK55FTe+kD4RbfMks1AHaqn3oKscUQ67FKRQaC8bdWr3ZoU1U9QY6
            file add dPj6JvFQml+cnclmhb5Qbui/WFU8F0hWB4kgpcs7VqGEkjAj0uHDNLkX8D41kkM=
            file end "-----END CERTIFICATE-----"
            file b64new SERVER_A.CER
            file add "-----BEGIN CERTIFICATE-----"
            file add MIIC2DCCAkGgAwIBAgIJANzplcbFEjVIMA0GCSqGSIb3DQEBBQUAMIGEMQswCQYD
            file add VQQGEwJFUzEPMA0GA1UECAwGTWFkcmlkMQ8wDQYDVQQHDAZNYWRyaWQxDzANBgNV
            file add BAoMBlRlbGRhdDEPMA0GA1UECwwGVGVsZGF0MQ8wDQYDVQQDDAZUZWxkYXQxIDAe
            file add BgkqhkiG9w0BCQEWEXRlbGRhdEB0ZWxkYXQuY29tMB4XDTE0MTAwODEwMDk1M1oX
            file add DTI0MTAwNTEwMDk1M1owgYQxCzAJBgNVBAYTAkVTMQ8wDQYDVQQIDAZNYWRyaWQx
            file add DzANBgNVBAcMBk1hZHJpZDEPMA0GA1UECgwGVGVsZGF0MQ8wDQYDVQQLDAZUZWxk
            file add YXQxDzANBgNVBAMMBlRlbGRhdDEgMB4GCSqGSIb3DQEJARYRdGVsZGF0QHRlbGRh
            file add dC5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBALoenHsvxFexXtKKtN0q
            file add fpYNo/t/epcsXIXIssKH6Eqq3YwMw1zjZfLhmi5W5MYqFUB/vW2fJHwArgb6oJk+
            file add 91230ipPrXSV5InMIeqw6JQMJb01RCWQCVVMu+2uAAeMP700dSIOwUnHfi/P3KmW
            file add pNS7adPQh10DdVumcv7Sn63zAgMBAAGjUDBOMB0GA1UdDgQWBBQ3vmphazbeQ8mK
            file add 5KlXa2rn1Ey8rDAfBgNVHSMEGDAWgBQ3vmphazbeQ8mK5KlXa2rn1Ey8rDAMBgNV
            file add HRMEBTADAQH/MA0GCSqGSIb3DQEBBQUAA4GBAKsRJRvTNVdJIxAvVdhPyF2GJZeZ
            file add hb3dBDZ11Bt145A3zof0zxaxBo4B/PnrHNeJ60I79uFBmNRQxAi7pa6J7ow+a0jq
            file add dOfPzmnv6Yz980OdWRmRbfbjC1mul3ZUTdXD+Sq8TX7cG46xqt23wuWHx1ql6tkg
            file add O5aHMHKpCrzyWCnq
            file end "-----END CERTIFICATE-----"
;
            certificate SERVER_A.CER load
            certificate SERVER.CER load
         exit
;
      exit
;
   exit
;
   protocol dhcp
; -- DHCP Configuration --
      server
; -- DHCP Server Configuration --
         enable
;
;
         shared 1 global-vrf
;
         subnet wlan 1 network 172.16.0.0 255.255.0.0
         subnet wlan 1 range 172.16.0.10 172.16.0.20
         subnet wlan 1 dns-server 192.168.212.3
         subnet wlan 1 router 172.16.0.1
;
      exit
;
   exit
;
;
;
   feature ntp
; -- NTP Protocol user configuration --
      protocol
      peer address 1 147.83.123.133
```

```
   exit
;
   feature radius
; -- RADIUS User Configuration --
      primary-address 172.17.0.253
      primary-secret my-radius-secret
      enable radius
   exit
;
   feature hotspot
; -- Hotspot Configuration --
      network wlan0/0
         redirect enable
         uam-server secure enable
         uam-server secure ca SERVER_A.CER
         uam-server secure user SERVER.CER
         url portal-page http://172.17.0.254/login_en.htm
         walled-garden access-list 101
         enable
      exit
;
   exit
;
   dump-command-errors
```

4.2 Example 2: captive portal in the cloud

The scenario for this example presents both AAA and captive portal servers in an external location in the cloud.

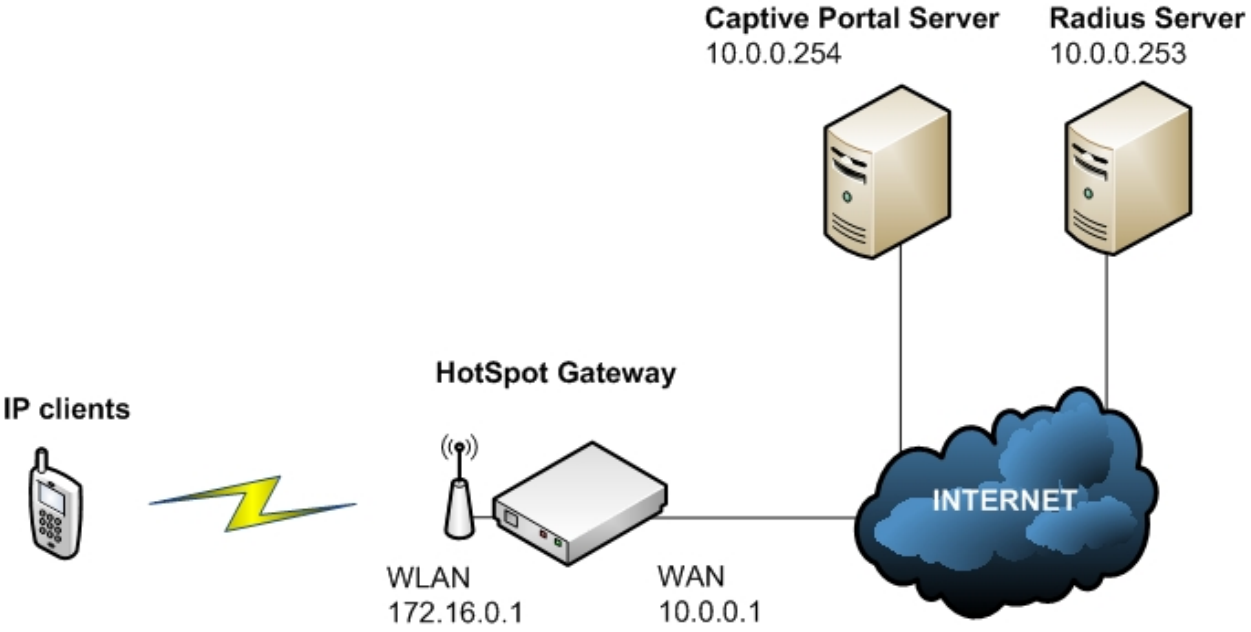

Fig. 5: **Example 2: Network diagram**

Wireless devices attempt to access the Internet through the WLAN interface (wlan0/0). To reach the Internet destinations, traffic is routed to the WAN interface (ethernet0/0). To obtain access, devices must be authenticated in the HotSpot. Servers used to authenticate clients must be reachable behind the WAN interface, which means a walled garden must be configured. The captive portal launches the authentication process for social networks and is set with a domain name.

4.2.1 AAA configuration

In this scenario, AAA services are also extended to execute subscriber session accounting. This function is provided by the AAA feature. Please see manual *Dm800-I AAA Feature* for further information.

Radius servers running in 10.0.0.253 are included in the *aaa-group* server group. Subscriber authentication is performed through the definition of an AAA **authentication login** method list.

To set specific session parameters per user, an AAA method list is defined: **authorization network.**

Finally, an AAA **accounting network** method list is defined for accounting purposes.

Configuration example:

```
   feature aaa
; -- AAA user configuration --
      enable
      radius-servers
         server "server-auth"
            port 1604
            key ciphered 0xFA06C224393809818E11055125E500A887A37B015D765AEE
            host 10.0.0.253
         exit
;
         server "server-acct"
            port 1605
            key ciphered 0xFA06C224393809818E11055125E500A887A37B015D765AEE
            host 10.0.0.253
         exit
;
      exit
;
      group server radius "aaa-group"
         server server-auth
         server server-acct
      exit
;
      authentication login "auth-list"
         method 1 group aaa-group
      exit
;
      authorization network "auth-list"
         method 1 group aaa-group
      exit
;
      accounting network "acct-list"
         action-type start-stop
         method 1 group aaa-group
      exit
;
   exit
```

4.2.2 Walled garden configuration

An access list must be configured to include *permit* entries. In this particular configuration example, all of them will be added as domains. The router will manage all DNS requests. The remaining packets must be classified as *non-accepted*, and be filtered when adding *deny* as a last entry.

Configuration example:

```
   feature access-lists
; -- Access Lists user configuration --
      access-list 101
         description "Walled Garden"
;
         entry 1 default
         entry 1 deny
;
      exit
;
   exit
```

4.2.3 HotSpot configuration

HotSpot configuration needs to redirect clients to the captive portal server running in 10.0.0.254. By configuring **redirect enable,** the *non-accepted* HTTP traffic is received by the UAM server, which processes the request and returns an HTTP response. The UAM server response redirects the HTTP request to the URL configured by the **url portal-page <URL text>** command. In this case, **url query-format loginurl** is used due to the specifications of the captive portal server. The walled garden configuration, using the previously defined access list, needs to let clients reach the captive portal server.

To enable AAA services in the HotSpot, AAA method lists (previously defined using the AAA feature) are configured. To enable periodic accounting record sending, enter **accounting interim-interval** .

To make social network authentication possible, a list of domains must be added to the **walled-garden** configuration (together with a walled garden access list). In this case, the captive portal implements authentication via *facebook* and *twitter*. Thus, all domains that take part in the authentication process must be added to the domain list.

In this example, user credentials are presented encrypted to the UAM server using a CHAP challenge and a shared secret between the UAM server and the captive portal. For this feature to work, the **uam-server authentication chap** and **uam-server authentication secret <secret>** commands need to be configured.

Finally, an access policy is configured in the HotSpot to discard all incoming packets classified as *non-accepted*, through the **policy drop** command.

Configuration example:

```
   feature hotspot
; -- Hotspot Configuration --
      network wlan0/0
         accounting interim-interval 1m
         accounting network acct-list
         authentication login auth-list
         authorization network auth-list
         policy drop
         redirect enable
         uam-server authentication chap
         uam-server authentication secret my-secret
         url portal-page http://10.0.0.254/login_en.htm
         url query-format loginurl
         walled-garden access-list 101
         walled-garden domain captive.portal.com
         walled-garden domain *facebook.es
         walled-garden domain *facebook.com
         walled-garden domain *facebook.net
         walled-garden domain *fbcdn.net
         walled-garden domain *akamaihd.net
         walled-garden domain ocsp.comodoca.com
         walled-garden domain *twitter.com
         walled-garden domain *twimg.com
         walled-garden domain *akamaiedge.net
         walled-garden domain ocsp.digicert.com

         enable
      exit
;
   exit
```

4.2.4 Complete configuration

A basic configuration for the proposed example is given below:

```
   log-command-errors
   no configuration
   feature access-lists
; -- Access Lists user configuration --
      access-list 101
         description "Walled Garden"
;
         entry 1 default
```

```
        entry 1 deny
;
      exit
;
   exit
;
   feature aaa
; -- AAA user configuration --
      enable
      radius-servers
         server "server-auth"
            port 1604
            key ciphered 0xFA06C224393809818E11055125E500A887A37B015D765AEE
            host 10.0.0.253
         exit
;
         server "server-acct"
            port 1605
            key ciphered 0xFA06C224393809818E11055125E500A887A37B015D765AEE
            host 10.0.0.253
         exit
;
      exit
;
      group server radius "aaa-group"
         server server-auth
         server server-acct
      exit
;
      authentication login "auth-list"
         method 1 group aaa-group
      exit
;
      authorization network "auth-list"
         method 1 group aaa-group
      exit
;
      accounting network "acct-list"
         action-type start-stop
         method 1 group aaa-group
      exit
;
   exit
;
;
   network ethernet0/0
; -- Ethernet Interface User Configuration --
      ip address 10.0.0.1 255.0.0.0
;
;
   exit
;
;
   network wlan0/0
; -- Wireless LAN Interface. Configuration --
      ip address 172.16.0.1 255.255.0.0
;
      bss "my-bss"
         privacy-invoked
         rsn wpa
         cipher aes-ccmp
         akm-suite psk
         wpa-psk passphrase ciphered 0xFCEFFD77D44110F0A8AA0C9B91146913
      exit
;
   exit
;
```

```
;
   protocol ip
; -- Internet protocol user configuration --
      route 0.0.0.0 0.0.0.0 10.10.10.254
;
      rule 1 local-ip ethernet0/0 remote-ip any
      rule 1 napt translation
;
   exit
;
   protocol dhcp
; -- DHCP Configuration --
      server
; -- DHCP Server Configuration --
         enable
;
         shared 1 global-vrf
;
         subnet wlan 1 network 172.16.0.0 255.255.0.0
         subnet wlan 1 range 172.16.0.10 172.16.0.20
         subnet wlan 1 dns-server 172.16.0.1
         subnet wlan 1 router 172.16.0.1
;
      exit
;
   exit
;
   feature dns
; -- DNS resolver user configuration --
      server 8.8.8.8
   exit
;
   feature ntp
; -- NTP Protocol user configuration --
      protocol
      peer address 1 147.83.123.133
   exit
;
   feature hotspot
; -- Hotspot Configuration --
      network wlan0/0
         accounting interim-interval 1m
         accounting network acct-list
         authentication login auth-list
         authorization network auth-list
         policy drop
         redirect enable
         session hard-timeout 1d
         session idle-timeout 5m
         uam-server authentication chap
         uam-server authentication secret my-secret
         url portal-page http://10.0.0.254/login_en.htm
         url query-format loginurl
         walled-garden access-list 101
         walled-garden domain captive.portal.com
         walled-garden domain *facebook.es
         walled-garden domain *facebook.com
         walled-garden domain *facebook.net
         walled-garden domain *fbcdn.net
         walled-garden domain *akamaihd.net
         walled-garden domain ocsp.comodoca.com
         walled-garden domain *twitter.com
         walled-garden domain *twimg.com
         walled-garden domain *akamaiedge.net
         walled-garden domain ocsp.digicert.com

         enable
```

```
      exit
;
   exit
;
   dump-command-errors
```

4.3 Example 3: using MAB and CoA functionalities

The scenario for this example presents both AAA and captive portal servers in an external location, reachable by the HotSpot Gateway through the WAN interface.

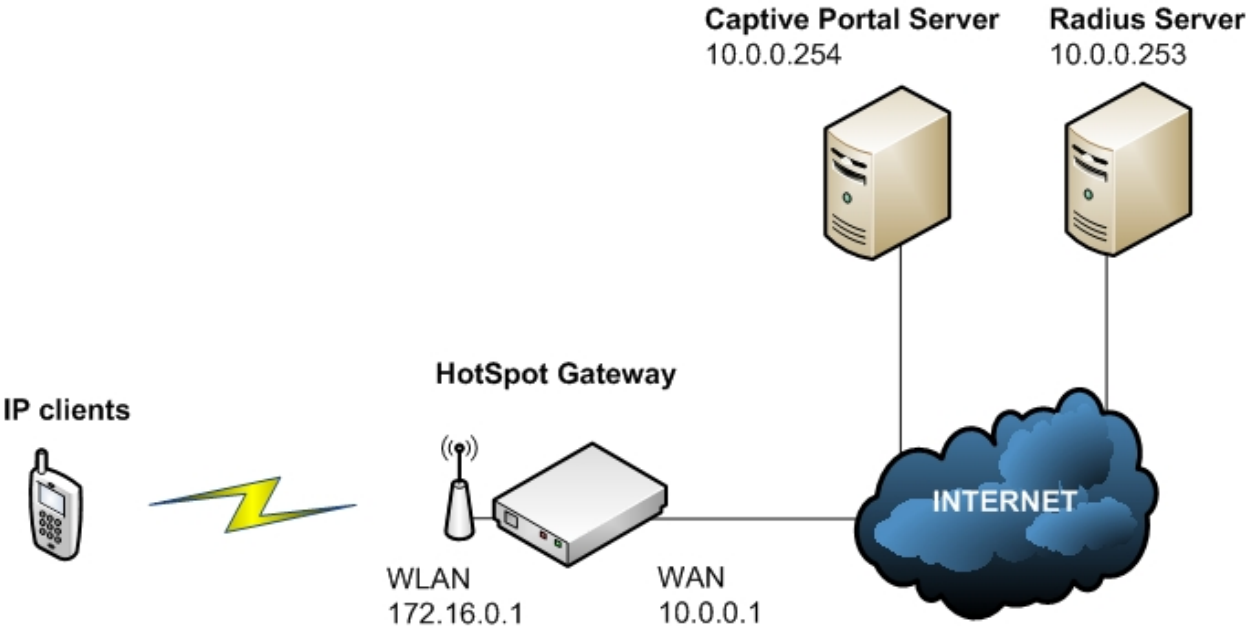

Fig. 6: **Example 3: Network diagram**

Wireless devices attempt to access the Internet through the WLAN interface (wlan0/0). To reach the Internet destinations, traffic is routed to the WAN interface (ethernet0/0). To be granted access, client devices must have an authorized session in the HotSpot feature. Remote servers used to authenticate users must be reachable by client devices beyond the WAN interface (meaning a walled garden must be properly configured).

4.3.1 AAA configuration

In this scenario, AAA services are configured to use MAC Authentication Bypass and Change of Authorization. The AAA feature provides these functionalities. Please see manual *Dm800-I AAA Feature* for further information.

A Radius server running in 10.0.0.253 is included in the *aaa-group* server group. MAC Authentication Bypass is performed by defining an AAA **authentication dot1x** method list.

To set specific session parameters per user, an AAA methods list is defined: **authorization network**.

A Radius server is configured to list Change of Authorization requests in the **change-of-authorization radius-server** submenu. Only requests from a Radius client located in 10.0.0.254 are allowed.

Configuration example:

```
    feature aaa
; -- AAA user configuration --
      enable
      radius-servers
         server "server-auth"
            key ciphered 0x1D198DA2EC44F099B575F3612529BF8EDE90653BEB92912D
            host 10.0.0.253
         exit
;
      exit
;
      group server radius "aaa-group"
         server server-auth
```

```
      exit
;
      authentication dot1x "auth-list"
         method 1 group aaa-group
      exit
;
      authorization network "auth-list"
         method 1 group aaa-group
      exit
;
      change-of-authorization radius-server
; -- CoA Radius user configuration --
         enable
         client "coa-auth"
            key ciphered 0x1D198DA2EC44F099B575F3612529BF8EDE90653BEB92912D
            host 10.0.0.254
         exit
;
      exit
;
   exit
```

4.3.2 Walled garden configuration

An access list must be configured to include *permit* entries for the captive portal server destination and DNS queries. The remaining packets are classified as *non-accepted* by adding a last *deny* entry.

Configuration example:

```
   feature access-lists
; -- Access Lists user configuration --
      access-list 101
         description "Walled Garden"
;
         entry 1 description "DNS Queries"
         entry 1 default
         entry 1 permit
         entry 1 destination port-range 53 53
         entry 1 protocol udp
;
         entry 2 description "Captive Portal Server"
         entry 2 default
         entry 2 permit
         entry 2 destination address 10.0.0.254 255.255.255.255
;
         entry 3 default
         entry 3 deny
;
      exit
```

4.3.3 HotSpot configuration

The HotSpot feature is configured so it redirects clients to the captive portal server running in 10.0.0.254. By configuring **redirect enable,** the *non-accepted* HTTP traffic is intercepted and received by the embedded UAM server. The UAM server responds by redirecting the client to the captive portal URL received in the MAB request response.

The MAB feature is enabled by the **mac-authentication-bypass enable** command. Also, the **mac-authentication-bypass activity-timeout** command is configured to set the time without retries for a rejected MAB request.

To perform MAB requests, the **authentication dot1x** command sets the necessary AAA method list.

Reception of CoA commands in the HotSpot feature is enabled by the **change-of-authorization enable** command.

Configuration example:

```
   feature hotspot
; -- Hotspot Configuration --
```

```
      network wlan0/0
;
         authentication dot1x auth-list
         authorization network auth-list
         change-of-authorization enable
         mac-authentication-bypass enable
         mac-authentication-bypass activity-timeout 5m
         policy drop
         redirect enable
         url query-format coa-ip 10.0.0.1
         url query-format coa-port 3799
         walled-garden access-list 101
         enable
      exit
;
   exit
```

4.3.4 Complete configuration

A basic configuration for the proposed example is given below:

```
   log-command-errors
   no configuration
   feature afs
      enable
   exit
;
   feature access-lists
; -- Access Lists user configuration --
      access-list 101
         description "Walled Garden"
;
         entry 1 description "DNS Queries"
         entry 1 default
         entry 1 permit
         entry 1 destination port-range 53 53
         entry 1 protocol udp
;
         entry 2 description "Captive Portal Server"
         entry 2 default
         entry 2 permit
         entry 2 destination address 10.0.0.254 255.255.255.255
;
         entry 3 default
         entry 3 deny
;
      exit
;
   exit
;
   feature aaa
; -- AAA user configuration --
      enable
      radius-servers
         server "server-auth"
            key ciphered 0x1D198DA2EC44F099B575F3612529BF8EDE90653BEB92912D
            host 10.0.0.253
         exit
;
      exit
;
      group server radius "aaa-group"
         server server-auth
      exit
;
      authentication dot1x "auth-list"
         method 1 group aaa-group
```

```
      exit
;
      authorization network "auth-list"
         method 1 group aaa-group
      exit
;
      change-of-authorization radius-server
; -- CoA Radius user configuration --
         enable
         client "coa-auth"
            key ciphered 0x1D198DA2EC44F099B575F3612529BF8EDE90653BEB92912D
            host 10.0.0.254
         exit
;
      exit
;
   exit
;
;
   network ethernet0/0
; -- Ethernet Interface User Configuration --
      ip address 10.0.0.1 255.0.0.0
;
   exit
;
;
   network wlan0/0
; -- Wireless LAN Interface. Configuration --
      ip address 172.16.0.1 255.255.0.0
;
      bss "my-bss"
         privacy-invoked
         rsn wpa
         cipher aes-ccmp
         akm-suite psk
         wpa-psk passphrase ciphered 0xFCEFFD77D44110F0A8AA0C9B91146913
      exit
;
   exit
;
   protocol ip
; -- Internet protocol user configuration --
      route 0.0.0.0 0.0.0.0 10.10.10.254
;
      nat
         rule 1 out ethernet0/0 dynamic overload
         rule 1 translation source interface ethernet0/0
;
      exit
;
   exit
;
   protocol dhcp
; -- DHCP Configuration --
      server
; -- DHCP Server Configuration --
         enable
;
         subnet wlan 1 network 172.16.0.0 255.255.0.0
         subnet wlan 1 range 172.16.0.10 172.16.0.20
         subnet wlan 1 dns-server 10.10.10.252
         subnet wlan 1 router 172.16.0.1
;
      exit
;
   exit
;
```

```
   feature hotspot
; -- Hotspot Configuration --
      network wlan0/0
;
         authentication dot1x auth-list
         authorization network auth-list
         change-of-authorization enable
         mac-authentication-bypass enable
         mac-authentication-bypass activity-timeout 5m
         policy drop
         redirect enable
         url query-format coa-ip 10.0.0.1
         url query-format coa-port 3799
         walled-garden access-list 101
         enable
      exit
;
   exit
;
   dump-command-errors
```

Chapter 5 Annex A

5.1 Third Party Software

When it comes to TLS negotiation, CIT uses the OpenSSL library code.

Please see a copy of the OpenSSL license below:

The OpenSSL toolkit remains under a dual license, i.e. both the conditions of the OpenSSL License and the original SSLeay license apply to the toolkit. The actual license texts can be found below.

OpenSSL License

Copyright (c) 1998-2019 The OpenSSL Project. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided the following conditions are met:

(1) Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

(2) Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

(3) All advertising materials mentioning features or use of this software must display the following acknowledgment: "This product includes software developed by the OpenSSL Project to be used in the OpenSSL Toolkit. (http://www.openssl.org/)"

(4) The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to endorse or promote products derived from this software without prior written permission. To obtain written permission, please contact openssl-core@openssl.org.

(5) Products derived from this software may not be called "OpenSSL" nor may "OpenSSL" appear in their names without the OpenSSL Project's prior written permission.

(6) Redistributions of any form whatsoever must retain the following acknowledgment: "This product includes software developed by the OpenSSL Project to be used in the OpenSSL Toolkit (http://www.openssl.org/)"

THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USAGE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) IN ANY WAY ARISING FROM THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com). This product includes software written by Tim Hudson (tjh@cryptsoft.com).

Original SSLeay License:

Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)

All rights reserved.

This package is an SSL implementation written by Eric Young (eay@cryptsoft.com).

The implementation was written so as to conform with Netscape's SSL.

This library is free for commercial and non-commercial use as long as the following conditions are adhered to. The following conditions apply to all code found in this distribution, be it the RC4, RSA, lhash, DES, etc., code; not just the SSL code. The SSL documentation included with this distribution is covered by the same copyright terms, save Tim Hudson (tjh@cryptsoft.com) is the holder.

Copyright remains Eric Young's, and as such any Copyright notices in the code are not to be removed. If this package is used in a product, Eric Young should be given attribution as the author of the parts of the library used. This can be in the form of a textual message at program startup or in documentation (online or textual) provided with the package.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the fol-

lowing conditions are met:

(1) Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.

(2) Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

(3) All advertising materials mentioning features or use of this software must display the following acknowledgement: "This product includes cryptographic software written by Eric Young (eay@cryptsoft.com)".
The word 'cryptographic' can be left out if the routines from the library being used are not cryptographically related.

(4) If you include any Windows specific code (or a derivative thereof) from the apps directory (application code) you must include an acknowledgement: "This product includes software written by Tim Hudson (tjh@cryptsoft.com)".

THIS SOFTWARE IS PROVIDED BY ERIC YOUNG "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) IN ANY WAY ARISING FROM THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The license and distribution terms for any publicly available version or derivative of this code cannot be changed. i.e. this code cannot simply be copied and put under another distribution license (including the GNU Public License).