AAA Feature

bintec-Dm 800

Legal Notice

Warranty

This publication is subject to change.

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

bintec 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

bintec-Dm704-I Configuration Monitoring

bintec-Dm710-I PPP Interface

bintec-Dm724-I FTP Protocol

bintec-Dm737-I HTTP Protocol

bintec-Dm738-I TELNET Protocol

bintec-Dm787-I SSH Protocol

bintec-Dm771-I Wireless LAN Interface

bintec-Dm783-I 802 1X MAB Authentication

Chapter 1 Introduction

1.1 Introduction to the AAA Feature

AAA (authentication, authorization and accounting) is a feature included in bintec routers that encompasses various protocols to manage the *authentication*, *authorization* and *accounting* processes. These processes are described below.

- *Authentication*: Process to identify a user who wishes to access the router. User identity is determined by a user ID, or login, and a password to access said router. Proof of identity is obtained by consulting a database, in either the local device or the remote server, which the router contacts through a secure protocol (such as Tacacs+).
- *Authorization*: Process to determine what resources a user can access in a system. Based on their identity, users are granted or denied access to said resources. Similarly to what happens with authentication, when a user wants to use a resource, his authorization is executed through a local database or one located in a remote security server.
- *Accounting*: Process to register the activity of various users in the router. Once processed, accounting information is particularly useful for administration and security tasks, etc.

The AAA feature responds to any of these three processes, robustly and independently. Robustness is rooted in the ability to resolve the same process through different protocols with backup alternatives.

The AAA feature works through the application of method lists. These define the method used in *authentication*, *authorization* and *accounting* processes. Said lists differ, depending on what they are used for: there are method lists, for example, for command authorization, user accounting, etc. When the system needs authenticating, authorizing or accounting, methods are applied in a pre-configured order.

The following figure shows a typical network scenario with an AAA configuration. This includes two servers for authentication/authorization, T1 and R1, and a Tacacs+ server for accounting, T2.

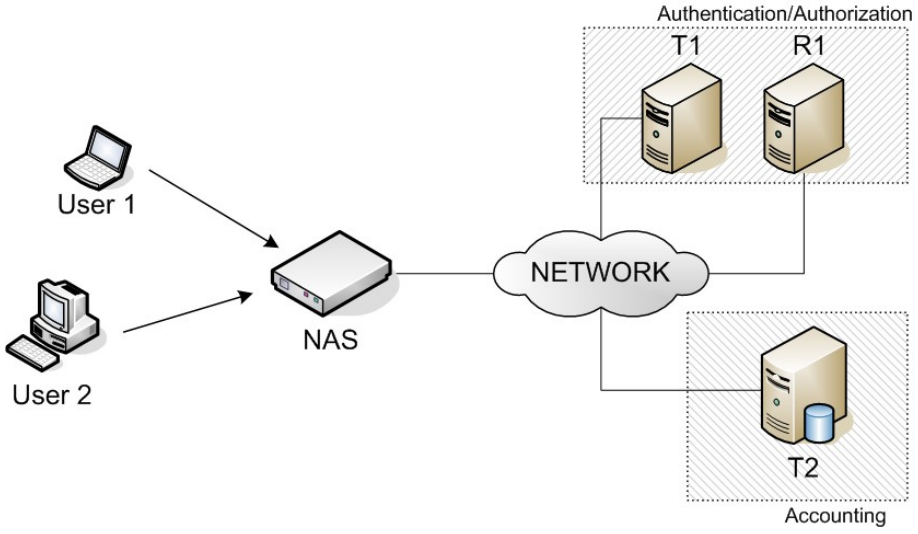

Fig. 1: Typical network configuration where AAA is used.

The network access server (NAS) initiates two authentication processes (the second one devoted to authorizing each user). Once the user accesses NAS, the latter registers his activity in the server and triggers accounting (T2).

When a user tries to access the router, the latter initiates an authentication process. At this point, the first method in the *authentication* method list is applied. In line with the scenario shown in Figure 1, this first method shows it must connect to the Tacacs+ (T1) server.

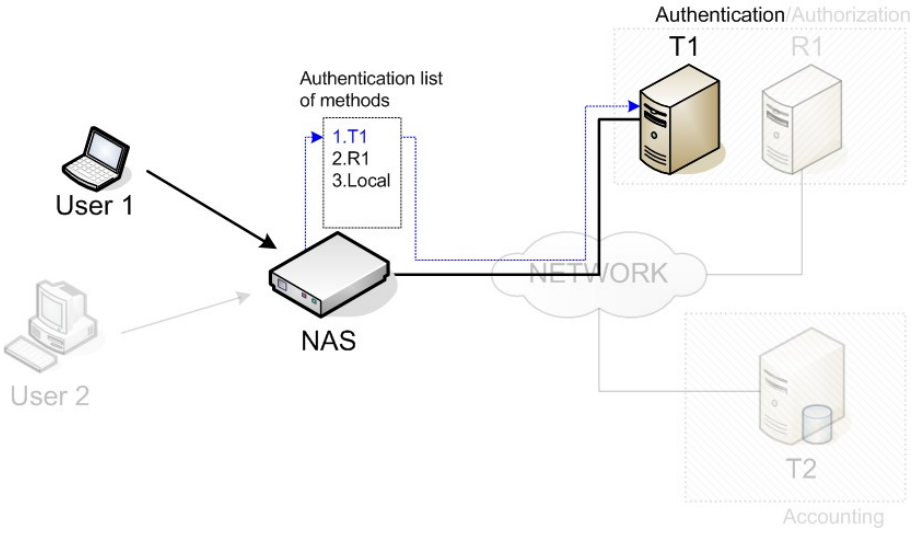

Fig. 2: Applying the first method from the authentication method list.

Occasionally, the server returns an error message. In this case it is considered down. Subsequently, the second method on the list is applied: this forwards a query to the Radius (R1) server.

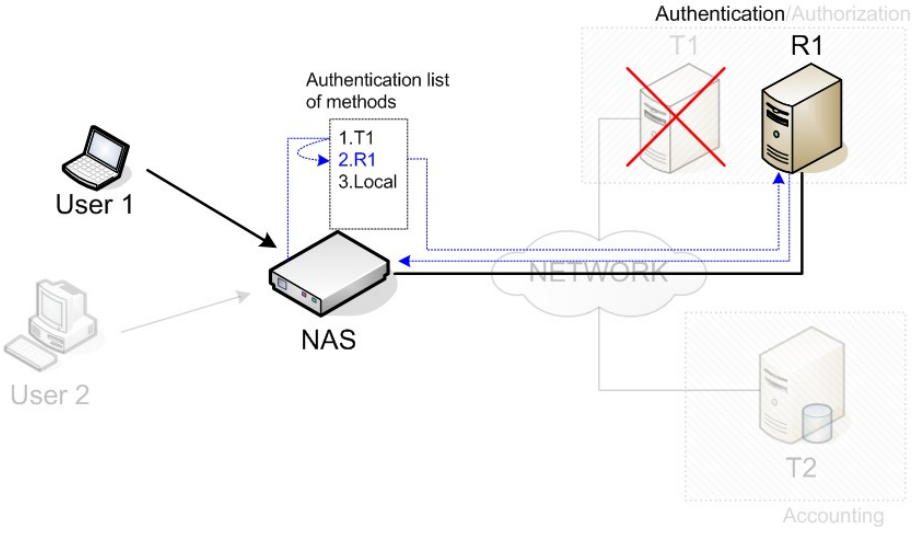

Fig. 3: Applying the second method from the authentication method list.

The R1 server resolves the authentication process and checks the user is who he says he is.

Once authentication has finished *authorization* starts. The method used is the same as above, but the list applied is the *authorization* list.

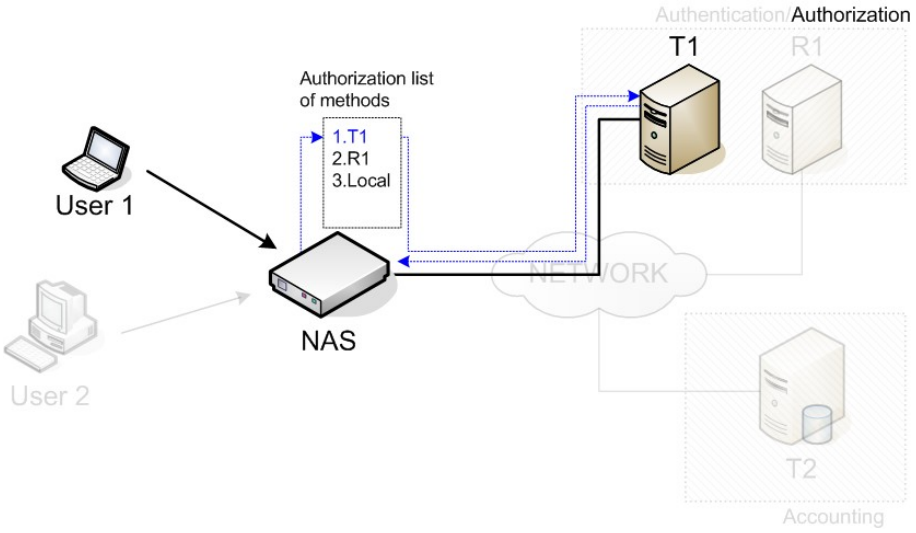

Fig. 4: Applying the first method on the authorization method list.

If the user succeeds in accessing the NAS, the latter begins to store user activity. The router begins an accounting

process each time an event requires registering. This process follows the method given by the *accounting* method list.

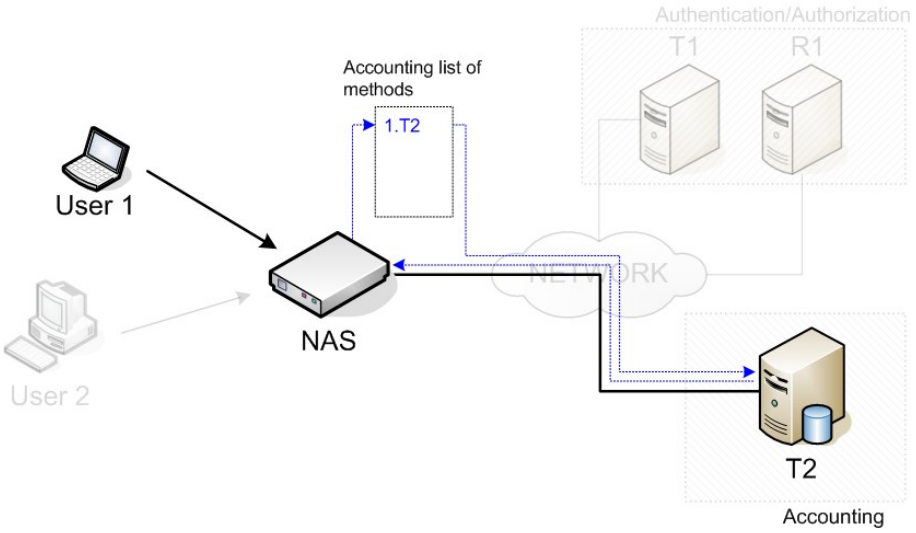

Fig. 5: Applying the first method on the accounting method list.

1.2 Change of Authorization

Change of Authorization is a functionality that provides a mechanism to dynamically perform changes over an AAA session after it is already authenticated. Those changes are requested by the administrator using some AAA protocol, and the NAS responds accepting or rejecting the requests.

For that purpose, the Radius extension standard for Dynamic Authorization is defined in the IETF RFC 5176. The Radius Dynamic Authorization can be used in the NAS by configuring the Change of Authorization functionality in the AAA feature.

In order to change an AAA session, the administrator must send a CoA request to the NAS. A CoA server must be listening in the NAS to receive those CoA requests and process them. The source of any CoA request is defined as a CoA client. The following packets performing CoA requests are defined in the Radius standard:

- Disconnect-Request: ends an AAA session. This means that the session switches from *authenticated* to *unauthenticated*. If the change is successfully completed, the NAS responds with a Disconnect-ACK packet. Otherwise, a Disconnect-NACK response that includes the *Error-Cause* Radius attribute and the error code is sent.
- CoA-Request: this packet is used to notify the NAS that some change on the session authorization must be done. The specific change needed is defined by the Radius attributes included in the request. If the change is successfully completed, the NAS responds with a CoA-ACK packet. Otherwise, the NAS sends a CoA-NACK response that includes the *Error-Cause* Radius attribute and the error code.

The IETF RFC 5176 defines the following error codes for the *Error-Cause* Radius attribute:

Error Code	Description
201	Residual Session Context Removed.
202	Invalid EAP Packet (Ignored).
401	Unsupported Attribute.
402	Missing Attribute.
403	NAS Identification Mismatch.
404	Invalid Request.
405	Unsupported Service.
406	Unsupported Extension.
407	Invalid Attribute Value.
501	Administratively Prohibited.
502	Request Not Routable (Proxy) .
503	Session Context Not Found.
504	Session Context Not Removable.
505	Other Proxy Processing Error.
506	Resources Unavailable.
507	Request Initiated.

508	Multiple Session Selection Unsupported.

Any CoA request must contain one or several Radius attributes, used to identify the AAA session in the NAS. The Radius attributes that help identify an AAA session in the NAS are the following:

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

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

In a CoA-Request packet, the Radius attributes included define the change to perform over an AAA session. The following Vendor-Specific Attributes (VSA) are provided for this purpose:

Attribute	**Value**	**Description**
Hotspot-Command	14	Integer that identifies the change to perform over an AAA session created in the HotSpot feature. This attribute can only be received in a CoA-Request packet.

The following are valid actions to perform using the *Hotspot-Command* vendor-specific attribute:

Command	**Value**	**Description**
Hotspot:Reauthenticate	0	Orders the HotSpot feature to request a new authentication for a given AAA session. If the AAA session is identified, the NAS responds with a CoA-ACK packet and proceeds to request the new authentication.

Chapter 2 Configuration

2.1 First Considerations

The AAA feature replaces old authentication and authorization systems. I.e. when the AAA feature is enabled, the router stops applying the configurations of these older systems. No access restrictions are applied in any of the services by default, except for those that need some parameters from the process like dot1x for wlan/ethernet interfaces, so you don't lose access to the router once the AAA feature is enabled. Any restrictions must be explicitly specified.

Note

By default, there are no router access restrictions for any service (except dot1x for wlan/ethernet interfaces, when AAA is enabled). They must be explicitly specified.

Please note that AAA is a tool that offers enormous control over router access and on-going events. However, poor configuration can result in the router becoming non-accessible. Knowing what to configure exactly is essential, as is ensuring the execution is carried out in a systematic and controlled manner.

Note

The AAA feature configuration must be executed in a systematic and controlled manner. A poorly configured AAA may mean losing access to the router.

Not all services support the same method lists (i.e. not all services offer the same AAA features). Table 1 shows the method lists supported by each service.

Methods

	Accounting			**Authentication**		**Authorization**		
	exec	commands	network	login	PPP	exec	commands	network
Console	✓	✓	n/a	✓	n/a	✓	✓	n/a
HTTP	n/a	n/a	n/a	✓	n/a	n/a	n/a	n/a
Telnet	✓	✓	n/a	✓	n/a	✓	✓	n/a
SSH	✓	✓	n/a	✓[1]	n/a	✓[2]	✓	n/a
FTP	✓	✗	n/a	✓	n/a	✓	✗	n/a
PPP	n/a	n/a	n/a	✓	✓	n/a	n/a	✓
HotSpot	n/a	n/a	✓	✓	n/a	n/a	n/a	n/a

✓: Supported.

✗: Not supported.

(N/A): Not applicable.

Similarly, not all parameters that AAA can assign (in the authorization process), are supported by all services. Table 2 shows the parameters supported by each service.

[1] Only available when SSH executes client authentication/authorization by means of a password or, as of version 11.01.10, a public key. For further information, please see manual *bintec-Dm787-I SSH Protocol*

[2] Only available when SSH executes client authentication/authorization by means of a password or, as of version 11.01.10, a public key. For further information, please see manual *bintec-Dm787-I SSH Protocol*

Authorization

	Maximum session time (Timeout)[3]	Maximum time without activity (Idletime)[4]	User privilege level[5]	IP address for the remote peer	Periodic accounting interval (interim-interval)
Console	T	T L	R T L	n/a	n/a
HTTP	n/a	n/a	n/a	n/a	n/a
Telnet	T	T L	R T L	n/a	n/a
SSH	✗	L	R T L	n/a	n/a
FTP	✗	L	R T L	n/a	n/a
PPP	n/a	n/a	n/a	R T L	n/a
HotSpot [6]	R	R	n/a	n/a	R

R: A Radius server can assign this parameter.

T: A Tacacs+ server can assign this parameter.

L: The local database can assign this parameter.

✗: This parameter isn't supported.

(N/A): This parameter doesn't apply to the service.

2.2 Configuring the AAA Feature: First Steps

The main goal of the AAA feature is to create and apply method lists (*local*, *none* or *group)*.

Group methods refer to a group of servers. There are groups of *Tacacs* + servers and *of Radius* servers.

Once method lists have been created, they can be applied to the different services. Consequently, a procedure for *authentication*, *authorization* and *accounting* can be set.

Figure 6 illustrates the above. The AAA configuration must be executed, according to the schema, from right to left (i.e. you must begin by configuring the servers and end by applying the method lists to the services).

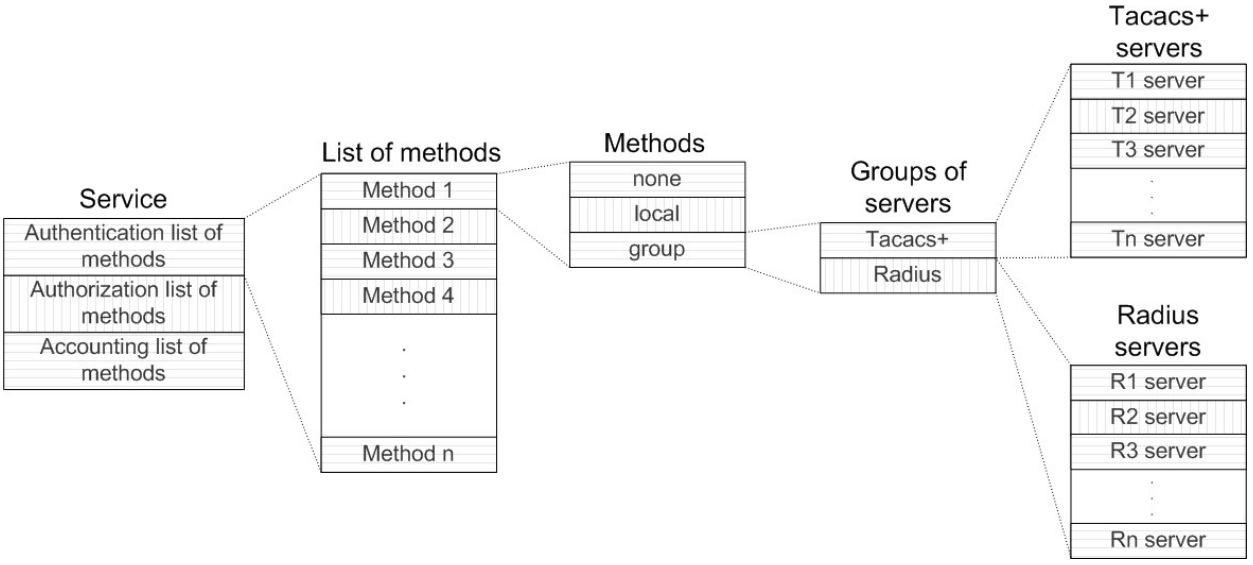

Fig. 6: Configuration schema for the AAA feature.

The following section shows you how to create the servers:

- RADIUS-SERVERS configuration menu commands, if you want to create Radius servers.
- TACACS-SERVERS configuration menu commands, if you want to create Tacacs+ servers.

The following sections focus on how to create group servers:

[3] If, during the *authorization* process, information from the *timeout* parameter isn't obtained, value 0 is set (i.e. no time limit). With the *hotspot* service, the feature's default configuration value is applied.
[4] If, during the *authorization* process, information from the *idletime* parameter isn't obtained, the value indicated in the local configuration is taken.
[5] If, during the authorization process, information from the user privilege level isn't obtained, a value of 15 is set (i.e. access as a *root* user).
[6] The hotspot service also supports a set of specific proprietary parameters for this feature. For more information, please see manual *bintec-Dm820-I Hotspot Feature*.

- GROUP SERVER RADIUS configuration menu commands, if you want to create a group of Radius servers.
- GROUP SERVER TACACS+ configuration menu commands, if you want to create a group of Tacacs+ servers.

The following sections explain, in greater detail, how to create the lists and their corresponding methods:

- ACCOUNTING COMMANDS configuration menu commands, if you want to register the commands executed in the router.
- ACCOUNTING EXEC configuration menu commands, if you want to register the number of accesses to the router.
- ACCOUNTING NETWORK configuration menu commands, if you want to register authorized network access through the router.
- AUTHENTICATION DOT1X configuration menu commands, if you want to execute authentication using 802.1X or using the source MAC addresses that access an interface.
- AUTHENTICATION LOGIN configuration menu commands, if you want to execute authentication for the users that access the router shell.
- AUTHENTICATION PPP configuration menu commands, if you want to execute authentication when establishing PPP connection in the router.
- AUTHORIZATION COMMANDS configuration menu commands, if you want to execute *authorization* to deny or allow commands to be executed.
- AUTHORIZATION EXEC configuration menu commands, if you want to execute *authorization* and impose parameters such as the privilege level for the users who access the router shell.
- AUTHORIZATION NETWORK configuration menu commands, if you want to execute *authorization* and impose parameters such as the remote IP address in PPP connections.

Section "Using the AAA feature in router services" covers how to apply method lists to several services. More information can be found in the manuals corresponding to each service.

- *bintec-Dm704-I Configuration Monitoring*, to apply the method lists in the console.
- *bintec-Dm737-I HTTP Protocol* , to apply the method lists in HTTP.
- *bintec-Dm738-I TELNET Protocol*, to apply the method lists in Telnet.
- *bintec-Dm787-I SSH Protocol* , to apply the method lists in SSH.
- *bintec-Dm724-I FTP Protocol*, to apply the method lists in FTP.
- *bintec-Dm710-I PPP Interface* , to apply the method lists in PPP.
- *bintec-Dm820-I Hotspot Feature*, to apply the method lists in the HotSpot functionality.
- *bintec-Dm771-I Wireless LAN Interface* , to apply the method lists in WLAN interfaces.
- *bintec-Dm783-I 802 1X MAB Authentication*, to apply the method lists in the Ethernet interfaces.

2.3 Accessing the Configuration

Access the AAA feature configuration menu through the main configuration menu (PROCESS 4) by executing the *feature aaa* command.

Syntax:

```
Config>feature aaa
-- AAA user configuration --
AAA config>?
  accounting                Accounting parameters
  authentication            Authentication parameters
  authorization             Authorization parameters
  change-of-authorization   Configure a Change of Authorization server
  enable                    Enables AAA subsystem
  group                     Configure a group
  no                        Negate a command or set its defaults
  radius-servers            Radius servers menu
  tacacs-servers            Tacacs+ servers menu
  exit                      Exit to parent menu
```

The following sections detail each command and what they are used for.

2.4 AAA Configuration Menu

The main AAA feature configuration menu contains the following commands:

Command	Function
? (HELP)	Displays the configuration commands or their options.
ACCOUNTING	Accesses the configuration for an *accounting* method list.
AUTHENTICATION	Accesses the configuration for an *authentication* method list.
AUTHORIZATION	Accesses the configuration for an *authorization* method list.
CHANGE-OF-AUTHORIZATION	Accesses the configuration for the *Change of Authorization* functionality.
ENABLE	Enables the AAA feature.
GROUP	Accesses the configuration for a group of servers.
NO	Negates a command or establishes its default parameters.
RADIUS-SERVER	Accesses the Radius servers' configuration menu.
TACACS-SERVER	Accesses the Tacacs+ servers' configuration menu.
EXIT	Returns to the configuration menu.

The following paragraphs describe each of the above commands.

2.4.1 ? (HELP)

Displays the available commands or their options.

2.4.2 ACCOUNTING

Accesses the configuration submenu for the *accounting* method lists.

Syntax:

```
AAA config>accounting {commands | exec | network} {default | <listname>}
```

commands:	The method list is *accountingcommands*.
exec:	The method list is *accountingexec*.
network:	The method list is *accounting network*.
default:	The method list is applied to all services with no explicitly defined features.
<listname>:	Method list's ID.

If the list with the specified identifier doesn't exist, it is created at this point. Three types of lists can be defined: *commands, exec* and *network*:

- *Commands* lists are used to register the commands executed in the router.
- *Exec* lists are used to register the users accessing the router.
- *Network* lists are used to register statistics on authorized network access through the router.

A method list with a *default* identifier is automatically applied to all services supporting this type of list. The default list is not applied when a different list for the service is explicitly specified.

Example 1:

```
AAA config>accounting exec AccExec
Exec list AccExec>
```

In example 1, you access the configuration menu for an *accounting exec* method list. If said list doesn't exist, one will be created on executing said command.

Example 2:

```
AAA config>accounting commands default
Cmds list default>
```

In example 2, we are defining an *accounting commands* method list that applies to all services supporting this and does not have an explicitly configured list.

Note

For further information, please see the sections on ACCOUNTING COMMANDS configuration commands, ACCOUNTING EXEC configuration commands and ACCOUNTING NETWORK configuration commands. Also, see the section on how to configure *accounting* method lists.

Command history:

Release	Modification
11.00.03	The **Accounting network** command has been introduced.

2.4.3 AUTHENTICATION

Accesses the configuration submenu for *authentication* method lists.

Syntax:

```
AAA config>authentication {dot1x | login | ppp} {default | <listname>}
```

dot1x	This is an *authentication dot1x* method list.
login	This is an *authentication login* method list.
ppp	This is an *authentication ppp* method list.
default	The method list is applied to all services with no explicitly defined features.
<listname>	Method list's ID.

If the list with the specified identifier doesn't exist, it is created at this point. The lists that can be defined are *login* and *ppp*:

- *dot1x* lists are used for authentication purposes using 802.1X or source MAC addresses able to access an interface.
- *login* lists are used to authenticate users accessing the router shell.
- *ppp* lists are used to authenticate PPP connections.

A method list with a *default* identifier is automatically applied to all services supporting this type of list. The default list is not applied when a different list for the service is explicitly specified.

Example 1:

```
AAA config>authentication login AutheLogin
Login list AutheLogin>
```

In example 1, we are accessing the configuration menu for a method list to authenticate the shell known as *AutheLogin*. If said list doesn't exist, one is created on executing this command.

Example 2:

```
AAA config>authentication ppp default
PPP list default>
```

In example 2, an *authentication* method list is defined and applied to all PPP interfaces without a configured list.

Note

For further information, please see the sections on AUTHENTICATION DOT1X configuration menu commands, AUTHENTICATION LOGIN configuration menu commands and AUTHENTICATION PPP configuration menu commands.

2.4.4 AUTHORIZATION

Use this command to access the configuration submenu for the *authorization* method lists.

Syntax:

```
AAA config>authorization {commands | exec | network} {default | <listname>}
```

commands	This is an *authorization commands* method list.
exec	This is an *authorization exec* method list.
network	This is an *authorization network* method list.
default	The method list is applied to all services that have no explicitly defined features.
<listname>	Method list's ID.

If the list with the specified identifier doesn't exist, it is created at this point. There are three types of *authorization*

lists: *commands, exec* and *network*:

- *Commands* lists determine what commands a user can execute at a given level.
- *Exec* lists determine the privileges a user has when accessing the router shell.
- *Network* lists determine the parameters that must be applied to a connection requesting *authorization* to access the network through the router, such as a PPP connection.

Method lists with a *default* identifier are automatically applied to all services supporting this type of lists. The default list is not applied when a different list is explicitly specified for the service.

Example:

```
AAA config>authorization exec AuthoExec
Exec list AuthoExec>
```

In this example, we are accessing the configuration menu for a method list to authorize the shell. If said list doesn't exist, one will be created on executing this command.

Note

For further information on how to configure the *authorization* method lists, please see the AUTHORIZATION COMMANDS configuration menu, AUTHORIZATION EXEC configuration menu and AUTHORIZATION NETWORK configuration menu.

Routers have 5 access levels defined in their default operating mode:

Numeric Value	Level
0	*None*
1	*Events*
5	*Monitor*
10	*Config*
15	*Root*

Table 3. Default access levels

User privileges authenticated in the router depend on the access level. For further information, please see manual *bintec-Dm704-I Configuration and Monitoring*.

2.4.5 CHANGE-OF-AUTHORIZATION

Accesses the configuration menu for the Change of Authorization (CoA) functionality.

Syntax:

```
AAA config>change-of-authorization radius-server
```

radius-server	Radius server for Change of Authorization.

Note

For further information, please see the CHANGE-OF-AUTHORIZATION RADIUS-SERVER configuration menu commands.

Command history

Release	Modification
11.00.06	The "*change-of-authorization radius-server*" command has been introduced as of version 11.00.06.
11.01.02	The "*change-of-authorization radius-server*" command has been introduced as of version 11.00.06.

2.4.6 ENABLE

Enables the AAA feature in the router.

Syntax:

```
AAA config>enable
```

Initially, the AAA feature is disabled.

The AAA feature manages, in the same way as the Radius feature does, the *authentication*/*authorization* for different services through the Radius protocol. Consequently, and to avoid collisions between the subsystems, disable the Radius feature before enabling AAA.

Example:

```
AAA config>enable
CLI Error: Radius is enabled. You must disable radius first
CLI Error: Command error
```

In this example, you can see an attempt to enable AAA when Radius is still enabled.

Note

We recommend that you enable the AAA feature once the configuration has been completed. If you don't, and the router hasn't been correctly configured, you may lose access to the router.

2.4.7 GROUP

Accesses the configuration menu for the server groups.

Syntax:

```
AAA config>group server {tacacs+ | radius} <groupname>
```

tacacs+	This is a group of Tacacs+ servers.
radius	This is a group of Radius servers.
<groupname>	Servers group identifier.

Servers can be grouped under the same name (through the servers' group identifier) for subsequent use. A server cannot be used if it hasn't been added to a group of servers.

Example 1:

```
AAA config>group server tacacs+ grupoTac
Tacacs+ group grupoTac>
```

In example 1, we are accessing the configuration menu for a Tacacs+ servers group. If said group doesn't exist, one is created on executing this command.

Example 2:

```
AAA config>group server radius grupoRad
Radius group grupoRad>
```

In example 2, we are accessing the configuration menu for a Radius servers group. If said group doesn't exist, one is created on executing this command.

Note

For further information, please see the sections on the GROUP SERVER TACACS+ configuration menu and the GROUP SERVER RADIUS configuration menu.

2.4.8 NO

Configures parameters using the default values or deletes the configuration.

Syntax:

```
AAA config>no accounting {commands | exec} <listname>
AAA config>no authentication {login | ppp} <listname>
AAA config>no authorization {commands | exec | network} <listname>
AAA config>no enable
AAA config>no group server {radius | tacacs+} <groupname>
```

Example:

```
AAA config>no group server tacacs+ grupoTac
AAA config>
```

In this example, a group of Tacacs+ servers is deleted.

2.4.9 RADIUS-SERVERS

Accesses the Radius servers' configuration submenu.

Syntax:

```
AAA config>radius-servers
Radius servers>
```

Note

For further information, please see the section on the RADIUS-SERVERS configuration menu.

2.4.10 TACACS-SERVERS

Accesses the Tacacs+ servers' configuration submenu.

Syntax:

```
AAA config>tacacs-servers
Tacacs+ servers>
```

Note

For further information, please see the section on the TACACS-SERVERS configuration menu.

2.5 ACCOUNTING COMMANDS Configuration Menu

In the ACCOUNTING COMMANDS configuration menu, you configure the method lists that deal with the accounting commands.

Note

The commands are accounted for, regardless of whether the execution procedure was successful or not.

The escape key and the exit command do not count.

Access this configuration menu from the AAA feature configuration menu.

```
AAA config>accounting commands AccCmds
Cmds list AccCmds>
```

In the first submenu, specify the level of commands you want to execute for accounting. Then access the ACCOUNTING COMMANDS configuration menu for a specific level of executed commands.

```
Cmds list AccCmds>privilege-level n
Cmds lvl n>
```

The following commands are available:

Command	Function
? (HELP)	Displays the configuration commands or their options.
ACTION-TYPE	Configures the accounting level.
METHOD	Defines a method within this method list.
NO	Negates a command or establishes its default parameters.
EXIT	Exits this configuration menu.

The following paragraphs describe each of the above commands.

2.5.1 ? (HELP)

Displays the available commands or their options.

2.5.2 ACTION-TYPE

Configures the level of detail under *accounting* for this method list. Default value is *start-stop.*

Syntax:

```
Cmds lvl n>action-type {none | start-stop | stop-only}
```

none:	The events aren't registered.
start-stop:	Start and stop events for each session are registered.
stop-only:	Only the stop events for the session are registered.

Example:

```
Cmds lvl 10>action-type stop-only
```

This example shows the method list must only register the stop event commands for the session.

Note

When executing commands, you can only carry out *accounting* at the end of a session. The same effect is achieved when you configure the action-type as start-stop or stop-only.

2.5.3 METHOD

Defines a method in the method list.

Syntax:

```
Cmds lvl n>method <priority> group <groupname>
```

<priority>:	Priority in the application of the method in the method list. The lower the priority, the quicker it's applied.
<groupname>:	Servers group identifier.

Radius server groups cannot be used (are not supported) in the *accounting commands* method lists.

Example:

```
Cmds lvl 10>method 1 group myGroup1
Cmds lvl 10>method 2 group myGroup2
Cmds lvl 10>
```

In this example, two methods are defined in the method list: a group of servers known as *myGroup1* and *myGroup2,* respectively. Server groups must be previously defined. For further information, please see the section on GROUP SERVER TACACS+ configuration menu.

Note

The router applies the next method on a list when the current method is a group of servers and a correct response hasn't been received from any server in the group.

2.5.4 NO

Configures parameters using the default values or deletes the configuration.

Syntax:

```
Cmds lvl n>no action-type
Cmds lvl n>no method <priority>
```

Example:

```
Cmds lvl 10>no method 1
Cmds lvl 10>
```

In this example, the method with priority 1 is eliminated.

2.6 ACCOUNTING EXEC Configuration Menu

In the ACCOUNTING EXEC configuration menu, you can configure the different method lists that apply for the accounting of router users' entries and exits.

This configuration menu is accessed from the AAA feature configuration menu.

```
AAA config>accounting exec AccExec
Exec list AccExec>
```

The following commands are available:

Command	Function
? (HELP)	Displays the configuration commands or their options.
ACTION-TYPE	Configures the accounting level.
METHOD	Defines a method within this method list.
NO	Negates a command or establishes its default parameters.
EXIT	Exits this configuration menu.

The following paragraphs describe each of the above commands.

2.6.1 ? (HELP)

Displays the available commands or their options.

2.6.2 ACTION-TYPE

Configures the level of detail under *accounting* for this method list. Default value is *start-stop.*

Syntax:

```
Exec list listname>action-type {none | start-stop | stop-only }
```

none:	Events aren't registered.
start-stop:	Start and stop events for each session are registered.
stop-only:	Only the stop events for the session are registered.

Example:

```
Exec list AccExec>action-type start-stop
```

This example shows that the method list must register both the start and end of a session (i.e. the service using this method list records the time a user enters and exits the device).

2.6.3 METHOD

Defines a method in the method list.

Syntax:

```
Exec list listname>method <priority> group <groupname>
```

<priority>:	Priority in the application of a method within the method list. The lower the priority, the quicker it's applied.
<groupname>:	Server group identifier.

Radius server groups cannot be used (are not supported) in the *accounting exec* method lists.

Example:

```
Exec list AccExec>method 1 group myGroup1
Exec list AccExec>method 2 group myGroup2
Exec list AccExec>
```

In this example, two methods are defined in the method list: two groups of servers known as *myGroup1* and *myGroup2* (respectively). Server groups must be previously defined. For further information, please see the section on GROUP SERVER TACACS+ configuration menu.

Note

The router applies the next method on a method list when the current method is a group of servers and a correct response hasn't been received from any server in the group.

2.6.4 NO

Configures parameters using the default values or deletes the configuration.

Syntax:

```
Exec list listname>no action-type
Exec list listname>no method <priority>
```

Example:

```
Exec list AccExec>no method 2
Exec list AccExec>
```

In this example, the method with priority 2 is eliminated.

2.7 ACCOUNTING NETWORK Configuration Menu

In the ACCOUNTING NETWORK configuration menu, you can configure the method lists responsible for recording statistics on connections that may grant network access through the router.

This configuration menu is accessed from the AAA Feature configuration menu.

```
AAA config>accounting network AccNet
Net list AccNet>
```

The following commands are available:

Command	Function
? (HELP)	Displays the configuration commands or their options.
ACTION-TYPE	Configures the accounting level.
METHOD	Defines a method within this method list.
NO	Negates a command or establishes its default parameters.
EXIT	Exits this configuration menu.

The following sections provide a detailed description of each of the above commands.

2.7.1 ? (HELP)

Displays the available commands or the options of one command.

Command history:

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

2.7.2 ACTION-TYPE

Configures the level of accounting detail for this method list. Its default value is *start-stop.*

Syntax:

```
Net list listname>action-type {none | start-stop | stop-only}
```

none:	No events are recorded.
start-stop:	Start and end of session events are recorded.
stop-only:	Only end-of-session events are recorded.

Example:

```
Net list AccNet>action-type start-stop
```

This example shows that the method list must register both the start and end of a session (i.e. the service using this method list records the time a user enters and exits the device).

Command history:

Release	Modification
11.00.03	The "*action-type*" command was introduced as of version 11.00.03

2.7.3 METHOD

Defines a method within the method list.

Syntax:

```
Net list listname>method <priority> group <groupname>
```

<priority>:	Priority in the application of the method within the method list. The lower the priority of the method, the earlier it is applied.
<groupname>:	Servers group name.

Accounting network in the method lists does not support the use of TACACS+ server groups.

Example:

```
Net list AccNet>method 1 group myGroup1
Net list AccNet>method 2 group myGroup2
Net list AccNet>
```

In this example two methods in the method list are defined: two server groups called *myGroup1* and *myGroup2*, respectively. The server groups must be previously defined. For further information, see the GROUP SERVER RADIUS configuration menu section.

Note

The router applies the next method on a list when the current method is a group of servers and it has not obtained the correct answer from any of the servers in the group.

Command history:

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

2.7.4 NO

Sets parameters with default values or deletes the configuration.

Syntax:

```
Net list listname>no action-type
Net list listname>no method <priority>
```

Example:

```
Net list AccNet>no method 2
Net list AccNet>
```

In the example, the method with priority 2 is removed.

Command history:

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

2.8 AUTHENTICATION DOT1X Configuration Menu

In the AUTHENTICATION DOT1X configuration menu, you can configure the method lists that execute *authentication* using 802.1X or the source MAC addresses that try to access an interface.

This configuration menu is accessed from the AAA feature configuration menu.

```
AAA config>authentication dop1x AutheDot1x
Dot1x list AutheDot1x>
```

The following commands are available:

Command	Function
? (HELP)	Displays the configuration commands or their options.
METHOD	Defines a method within this method list.
NO	Negates a command or establishes its default parameters.
EXIT	Exits this configuration menu.

The following paragraphs describe each of the above commands.

2.8.1 ? (HELP)

Displays the available commands or their options.

Command history

Release	Modification
11.00.06	The "*help*" command was introduced as of version 11.00.06.
11.01.02	The "*help*" command was introduced as of version 11.01.02.

2.8.2 METHOD

Defines a method in the method list.

Syntax:

```
Dot1x list listname>method <priority> {group <groupname> | none}
```

<priority>	Priority in the application of the method in the method list. The lower the priority, the quicker it's applied.
group:	The method is group. This uses a servers group.
<groupname>	Servers group identifier.
none	The method is none. This doesn't require authentication.

In *authentication dot1x* method lists, you can establish two types of methods:

- *group*: dot1x authentication is executed using a group of servers. The use of Tacacs+ servers groups in *authentication dot1x* is not supported.
- *none*: authentication isn't required. WLAN interfaces do not support this method.

Example:

```
Dot1x list AutheDot1x>method 1 group myGroup2
Dot1x list AutheDot1x>method 2 none
Dot1x list AutheDot1x>
```

In this example, two methods are defined in the method list. The first refers to a group of servers known as *myGroup2* and the second refers to "no authentication required" (i.e. if a correct response hasn't been received from any server belonging to *myGroup2*, authentication is considered successful).

The servers group must be previously defined. For further information, please see the section on the GROUP SERVER RADIUS configuration menu.

Note

The router applies the next method on a list when the current method is a group of servers and it has not obtained the correct answer from any of the servers in the group. If there is no next method, access is denied.

Command history

Release	Modification
11.00.06	The "*method*" command was introduced as of version 11.00.06.
11.01.02	The "*method*" command was introduced as of version 11.01.02.
11.01.08	The "*none*" method is supported by Ethernet interfaces.

2.8.3 NO

Configures parameters using the default values or deletes the configuration.

Syntax:

```
Dot1x list listname>no method <priority>
```

Example:

```
Dot1x list AutheDot1x>no method 1
Dot1x list AutheDot1x>
```

In this example, the method with priority 1 is eliminated.

Command history

Release	Modification
11.00.06	The "*no*" command was introduced as of version 11.00.06.
11.01.02	The "*no*" command was introduced as of version 11.01.02.

2.9 AUTHENTICATION LOGIN Configuration Menu Commands

In the AUTHENTICATION LOGIN configuration menu, you can configure the method lists applied when authenticating users who access the router shell.

This configuration menu is accessed from the AAA feature configuration menu.

```
AAA config>authentication login AutheLogin
Login list AutheLogin>
```

The following commands are available:

Command	Function
? (HELP)	Displays the configuration commands or their options.
METHOD	Defines a method within this method list.
NO	Negates a command or establishes its default parameters.
EXIT	Exits this configuration menu.

The following paragraphs describe each of the above commands.

2.9.1 ? (HELP)

Displays the available commands or their options.

2.9.2 METHOD

Defines a method within the method list.

Syntax:

```
Login list listname>method <priority> {group <groupname> | local | none}
```

<priority>:	Priority in the application of the method in the method list. The lower the priority, the quicker it's applied.
group:	The method is group, meaning a servers group is used.
<groupname>:	Servers group identifier.
local:	The method is local. This uses the local database.
none:	The method is none. This doesn't require authentication.

In *authentication login* method lists you can establish three types of methods:

- *group*: user authentication is executed using a group of servers.
- *local*: the user authenticates using the router's local database.
- *none*: authentication isn't required.

Example:

```
Login list AutheLogin>method 1 group myGroup1
Login list AutheLogin>method 2 local
Login list AutheLogin>
```

In this example, two methods are defined in the method list. The first refers to a group of servers known as *myGroup1* and the second to the local database.

The servers group must be previously defined. (Please see the section on the GROUP SERVER TACACS+ configuration menu and the GROUP SERVER RADIUS configuration menu for further information).

Note

The router applies the next method on a list when the current method is a group of servers and it has not obtained the correct answer from any of the servers in the group. If there is no next method, access is denied.

2.9.3 NO

Configures parameters using the default values or deletes the configuration.

Syntax:

```
Login list listname>no method <priority>
```

Example:

```
Login list AutheLogin>no method 1
Login list AutheLogin>
```

In this example, the method with priority 1 is eliminated.

2.10 AUTHENTICATION PPP Configuration Menu

In the AUTHENTICATION PPP configuration menu, you can configure the method lists that execute *authentication* in the PPP links.

This configuration menu is accessed from the AAA feature configuration menu.

```
AAA config>authentication ppp AuthePPP
PPP list AuthePPP>
```

The following commands are available:

Command	Function
? (HELP)	Displays the configuration commands or their options.
METHOD	Defines a method within this method list.
NO	Negates a command or establishes its default parameters.
EXIT	Exits this configuration menu.

The following paragraphs describe each of the above commands.

2.10.1 ? (HELP)

Displays the available commands or their options.

2.10.2 METHOD

Defines a method within the method list.

Syntax:

```
PPP list listname>method <priority> {group <groupname> | local | none}
```

<priority>:	Priority in the application of the method within the method list. The lower the priority, the quicker it's applied.
group:	The method is group, meaning a servers group is used.
<groupname>:	Servers group identifier.
local:	The method is local, meaning the local database is used.
none:	The method is none, meaning authentication is not required.

In *authentication ppp* method lists, you can establish three types of methods:

- *group*: user authentication is executed using a group of servers.
- *local*: authentication is executed using the router's local database.
- *none*: authentication isn't required.

Example:

```
PPP list AuthePPP>method 1 group myGroup2
PPP list AuthePPP>method 2 none
PPP list AuthePPP>
```

In this example, two methods are defined in the method list. The first refers to a group of servers known as *myGroup2* and the second refers to "no authentication required" (i.e. if a correct response hasn't been received from any server belonging to *myGroup2*, authentication is considered successful).

The servers group must be previously defined. Please see the section on the GROUP SERVER TACACS+ configuration menu and the GROUP SERVER RADIUS configuration menu for further information.

Note

The router applies the next method on a list when the current method is a group of servers and it has not obtained the correct answer from any of the servers in the group. If there is no next method, access is denied.

2.10.3 NO

Configures parameters using the default values or deletes the configuration.

Syntax:

```
PPP list listname>no method <priority>
```

Example:

```
PPP list AuthePPP>no method 1
PPP list AuthePPP>
```

In this example, the method with priority 1 is eliminated.

2.11 AUTHORIZATION COMMANDS Configuration Menu

In the AUTHORIZATION COMMANDS configuration menu, you can configure the method lists that authorize the commands executed in the router.

Note

The exit command and the escape key always execute and never start the *authorization* process

This configuration menu is accessed from the AAA feature configuration menu.

```
AAA config>authorization commands AuthorCmds
Cmds list AuthorCmds>
```

From this first submenu, you must first specify what level of commands you want to execute *authorization* for. Afterwards, access the AUTHORIZATION COMMANDS configuration menu for a certain level of executed commands.

```
Cmds list AuthorCmds>privilege-level n
Cmds lvl n>
```

The following commands are available:

Command	Function
? (HELP)	Displays the configuration commands or their options.
METHOD	Defines a method within the method list.
NO	Negates a command or establishes its default parameters.
USE-COMMAND-PATH	Uses the complete command path.
EXIT	Exits this configuration menu.

The following paragraphs describe each of the above commands.

2.11.1 ? (HELP)

Displays the available commands or their options.

2.11.2 METHOD

Defines a method in the method list.

Syntax:

```
Cmds lvl n>method <priority> {group <groupname> | local | none}
```

<priority>:	Priority in the application of the method within the method list. The lower the priority, the quicker it's applied.
group:	The method is group, meaning a servers group is used. The use of Radius servers groups in *authorization commands* is not supported.
<groupname>:	Servers group identifier.
local:	The method is local, meaning the local database is used.
none:	The method is none, meaning *authorization* isn't required.

In *authorization commands* method lists, you can establish three types of methods:

- *group: authorization* is executed using a group of servers. The use of Radius server groups in *authorization commands* is not supported.
- *local*: the local database rules whether the user can execute the commands. In this case, the user is authorized when he is registered or when the local database does not apply any kind of restriction (i.e. when it does not contain a user).
- *none*: authentication isn't required.

Example:

```
Cmds lvl 10>method 1 group myGroup1
Cmds lvl 10>method 2 none
Cmds lvl 10>
```

In this example, two methods are defined in the method list. The first refers to a group of servers known as *myGroup1* and the second refers to "no authentication required" (i.e. if a correct response hasn't been received from any server belonging to *myGroup1*, authentication is considered successful).

The servers group must be previously defined. For further information, please see the GROUP SERVER TACACS+ configuration menu section.

Note

The router applies the next method on a list when the current method is a group of servers and it has not obtained the correct answer from any of the servers in the group. If there is no next method, access is denied.

2.11.3 NO

Configures parameters with the default values or deletes the configuration.

Syntax:

```
Cmds lvl n>no method <priority>
```

Example:

```
Cmds lvl 10>no method 1
Cmds lvl 10>
```

In this example, the method with priority 1 is eliminated.

2.11.4 USE-COMMAND-PATH

The complete command path must be used in the authentication process.

Note

The complete path is only used when *authorization* is executed with a Tacacs+ server. Otherwise, this does not apply.

Syntax:

```
Cmds lvl n>use-command-path
```

By default, the complete command path is not used.

Example:

```
AAA config>authorization commands PathList
Cmds list PathList>privilege-level 1
Cmds lvl 1>use-command-path
Cmds lvl 1>
```

In this example, level 1 privilege is configured from the *PathList* method list so the complete command path is used in the authentication process.

If this option is enabled, you need to bear in mind that the command sent to the Tacacs+ server includes the complete path. The command is configured following these rules:

(1) The command begins with the > character.

(2) The command begins with >*monitor*> if it is a command from the monitoring menu.

(3) The command begins with >*config*> if it is a command from the configuration menu, regardless of whether it's static or dynamic.

(4) The different levels/menus are separated with the > character.

(5) Spaces are replaced by the period character (.).

(6) If the command is more than 255 characters long, it is shortened by eliminating the final characters.

Example 1:

```
>monitor>protocol.ip>ipsec>clear
```

This example shows how to send the *clear* command from the *IPSec* monitoring menu.

Example 2:

```
>monitor>feature.dns-updater>clear
```

This example shows how to send the *clear* command from the *dns-updater* monitoring menu.

Example 3:

```
>config>feature.aaa>enable
```

This example shows how to send the *enable* command from the *AAA* feature configuration menu.

Example 4:

```
>config>protocol.ip>ipsec>enable
```

This example shows how to send the *enable* command from the *IPSec* configuration menu.

Example 5:

```
>view
```

This example shows how to send the *view* command.

2.12 AUTHORIZATION EXEC Configuration Menu

You can configure the method lists used to establish the privileges for the users who access the router shell in the AUTHORIZATION EXEC configuration menu.

If you use a Radius server as an *authorization* method, the petition is forwarded with the word bintec in the password field. If the Radius server is already being used in authentication, then a second petition is not executed (as the information received in the first petition is being used).

The following parameters can be established in the *authorization* process, depending on the service and methods used:

- The user privilege level.
- The maximum time the shell is assigned to the user.
- The maximum time the user can remain inactive.

This configuration menu is accessed from the AAA feature configuration menu.

```
AAA config>authorization exec AuthorExec
Exec list AuthorExec>
```

The following commands are available:

Command	Function
? (HELP)	Displays the configuration commands or their options.
METHOD	Defines a method within the method list.
NO	Negates a command or establishes its default parameters.
EXIT	Exits the configuration menu.

The following paragraphs describe each of the above commands.

2.12.1 ? (HELP)

Displays the available commands or their options.

2.12.2 METHOD

Defines a method within the method list.

Syntax:

```
Exec list listname>method <priority> {group <groupname> | local | none}
```

<priority>:	Priority in the application of the method within the method list. The lower the priority, the quicker it's applied.
group:	The method is group, meaning a servers group is used.
<groupname>:	Servers group identifier.
local:	The method is local, meaning the local database is used.
none:	The method is none, meaning authentication is not required.

In *authorization exec* method lists, you can establish three types of methods:

- *group*: *authorization* is executed using a group of servers.
- *local*: *authorization* is executing using the router's local database.
- *none*: *authorization* isn't required.

Example:

```
Exec list AuthorExec>method 1 group myGroup1
Exec list AuthorExec>method 2 local
Exec list AuthorExec>
```

In this example, two methods are defined in the method list. The first refers to a group of servers known as *myGroup1* and the second refers to the local database (i.e. if a correct response hasn't been received from any server belonging to *myGroup1*, then *authorization* is executed using the router's internal information).

The servers group must be previously defined. Please see the section on the GROUP SERVER TACACS+ configuration menu and the GROUP SERVER RADIUS configuration menu for further information.

Note

The router applies the next method on a list when the current method is a group of servers and it has not obtained the correct answer from any of the servers in the group. If there is no next method, access is denied.

2.12.3 NO

Configures parameters using the default values or deletes the configuration.

Syntax:

```
Exec list listname>no method <priority>
```

Example:

```
Exec list AuthorExec>no method 1
Exec list AuthorExec>
```

In this example, the method with priority 1 is eliminated.

2.13 AUTHORIZATION NETWORK Configuration Menu Commands

In the AUTHORIZATION NETWORK configuration menu, you can configure the method lists responsible for authorizing connections that grant network access through the router (such as PPP links).

If you use a Radius server as an *authorization* method, the petition is forwarded with the word bintec in the password field. If the Radius server is already being used in authentication, then a second petition is not executed (as the information received in the first petition is being used).

You can establish the IP address for the peer in the *authorization* process.

This configuration menu is accessed from the AAA feature configuration menu.

```
AAA config>authorization network AuthorNet
Net list AuthorNet>
```

The following commands are available:

Command	Function
? (HELP)	Displays the configuration commands or their options.
METHOD	Defines a method within the method list.
NO	Negates a command or establishes its default parameters.
EXIT	Exits the configuration menu.

The following paragraphs describe each of the above commands.

2.13.1 ? (HELP)

Displays the available commands or their options.

2.13.2 METHOD

Defines a method within the method list.

Syntax:

```
Net list listname>method <priority> {group <groupname> | local | none}
```

<priority>:	Priority in the application of the method within the method list. The lower the priority, the quicker it's applied.
group:	The method is group. This uses a servers group.
<groupname>:	Servers group identifier.
local:	The method is local. This uses the local database.
none:	The method is none. This doesn't require authentication.

In *authorization network* method lists you can establish three types of methods:

- *group*: *authorization* is executed using a group of servers.
- *local*: *authorization* is executed using the router's local database.
- *none*: *authorization* isn't required.

Example:

```
Net list AuthorNet>method 1 group myGroup1
Net list AuthorNet>method 2 none
Net list AuthorNet>
```

In this example, two methods are defined in the method list. The first refers to a group of servers known as *myGroup1* and the second refers to "no *authorization* required" (i.e. if a correct response hasn't been received from any server belonging to *myGroup1*, then *authorization* is considered successful).

The servers group must be previously defined. For further information, please see the section on the GROUP SERVER TACACS+ configuration menu and the GROUP SERVER RADIUS configuration menu.

Note

The router applies the next method on a list when the current method is a group of servers and it has not obtained the correct answer from any of the servers in the group. If there is no next method, *authorization* is denied.

2.13.3 NO

Configures parameters using the default values or deletes the configuration.

Syntax:

```
Net list listname>no method <priority>
```

Example:

```
Net list AuthorNet>no method 1
Net list AuthorNet>
```

In this example, the method with priority 1 is eliminated.

2.14 CHANGE-OF-AUTHORIZATION RADIUS-SERVER Configuration Menu

The Radius Change of Authorization (CoA) functionality in the AAA feature is configured in the CHANGE-OF-AUTHORIZATION RADIUS-SERVER configuration menu.

The Radius CoA root configuration menu is accessed from the AAA feature configuration menu.

```
AAA config>change-of-authorization radius-server
CoA Radius>
```

The following commands are available:

Command	Function
? (HELP)	Displays the configuration commands or their options.
CLIENT	Accesses the configuration menu for a CoA Radius client.
ENABLE	Enables the CoA Radius server.
KEY	Establishes the key for the CoA Radius clients, which take the default value in this field.
NO	Negates a command or establishes its default parameters.
PORT	Establishes the CoA Radius server port.
EXIT	Exits the configuration menu.

The purpose of the commands to configure CoA clients is to simplify and speed up the client's configuration. The changes made using those commands affect all the CoA Radius clients with default values. For instance, if you configure "key plain 1234" in this menu, then all the clients with a default key will use value 1234.

2.14.1 ? (HELP)

Displays the available commands or their options.

Command history

Release	Modification
11.00.06	The "*help*" command was introduced as of version 11.00.06.
11.01.02	The "*help*" command was introduced as of version 11.01.02.

2.14.2 CLIENT

Accesses the configuration menu for an individual Radius Change of Authorization (CoA) client.

Syntax:

```
CoA Radius>client <client_name>
```

<client_name>	Name used to identify the CoA client.

Example:

```
CoA Radius>client "my-client1"
Radius client my-client1>
```

In this example, the client's identifier is defined as "my-client1".

In the configuration menu for an individual client, the following commands are available:

Command	Function
? (HELP)	Displays the configuration commands or their options.
HOST	Establishes the IP address for this CoA client.
KEY	Establishes the key for this CoA client.
NO	Negates a command or establishes its default parameters.
EXIT	Exits this configuration menu.

The following paragraphs describe each of the above commands.

2.14.2.1 ? (HELP)

Displays the available commands or their options.

Command history

Release	Modification
11.00.06	The "*help*" command was introduced as of version 11.00.06.
11.01.02	The "*help*" command was introduced as of version 11.01.02.

2.14.2.2 HOST

Defines the IP address for the CoA Radius client.

Syntax:

```
Radius client clientname>host <ipaddress> [vrf <vrf_name>]
```

<ipaddress>:	Client IP address.
<vrf_name>:	VRF used to communicate with the client.

Example:

```
Radius client my-client1>host 2.2.2.2
Radius client my-client1>
```

This example defines the client's IP address as 2.2.2.2.

Command history

Release	Modification
11.00.06	The "*host*" command was introduced as of version 11.00.06.
11.01.02	The "*host*" command was introduced as of version 11.01.02.

2.14.2.3 KEY

Defines the key to exchange traffic with this client.

Syntax:

```
Radius client clientname>key {plain | ciphered} <key>
```

plain:	This key is not encoded.
ciphered:	This key is encrypted.
<key>:	Client's key.

The *plain* option specifies a plain key and the *ciphered* option specifies an encrypted key.

Example 1:

```
Radius client my-client1>key plain 1234
Radius client my-client1>
```

Example 1 sets a plain key.

Example 2:

```
Radius client my-client1>key ciphered 0x4698601DE5BFA77D
Radius client my-client1>
```

Example 2 sets an encrypted key.

Note

If the key takes the default value, it takes it from the root menu configuration located in the Change-of-Authorization Radius Server configuration.

Command history

Release	Modification
11.00.06	The "*key*" command was introduced as of version 11.00.06.
11.01.02	The "*key*" command was introduced as of version 11.01.02.

2.14.2.4 NO

Configures parameters using the default values or deletes the configuration.

Syntax:

```
Radius client clientname>no host
Radius client clientname>no key
```

Example:

```
Radius client my-client1>no host
Radius client my-client1>
```

This example deletes the host IP address of the CoA client.

Command history

Release	Modification
11.00.06	The "*no*" command was introduced as of version 11.00.06.
11.01.02	The "*no*" command was introduced as of version 11.01.02.

2.14.3 ENABLE

Enables the Radius Change of Authorization (CoA) server.

Syntax:

```
CoA Radius>enable
```

Command history

Release	Modification
11.00.06	The "*enable*" command was introduced as of version 11.00.06.
11.01.02	The "*enable*" command was introduced as of version 11.01.02.

2.14.4 KEY

Defines the key to exchange traffic with the Radius Change of Authorization (CoA) clients that have a configured default key.

Syntax:

```
CoA Radius>key {plain | ciphered} <key>
```

plain:	This key is not encoded.
ciphered:	This key is encrypted.
<key>:	Server key.

The *plain* option specifies a plain key and the *ciphered* option specifies an encrypted key.

Example 1:

```
CoA Radius>key plain 1234
CoA Radius>
```

Example 1 sets a plain key.

Example 2:

```
CoA Radius>key ciphered 0x4698601DE5BFA77D
CoA Radius>
```

Example 2 sets an encrypted key.

Command history

Release	Modification
11.00.06	The "*key*" command was introduced as of version 11.00.06.
11.01.02	The "*key*" command was introduced as of version 11.01.02.

2.14.5 NO

Configures parameters using the default values or deletes the configuration.

Syntax:

```
CoA Radius>no client
CoA Radius>no enable
CoA Radius>no key
CoA Radius>no port
```

Example:

```
CoA Radius>no port
```

This example establishes timeout at its default value.

Command history

Release	Modification
11.00.06,	The "*no*" command was introduced as of version 11.00.06.
11.01.02	The "*no*" command was introduced as of version 11.01.02.

2.14.6 PORT

Establishes the Radius Change of Authorization (CoA) server port. Default is 3799.

Syntax:

```
CoA Radius>port <port>
```

<port>	Port.

Example:

```
CoA Radius>port 5000
CoA Radius>
```

This example sets the port to 5000.

Command history

Release	Modification
11.00.06	The "*port*" command was introduced as of version 11.00.06.
11.01.02	The "*port*" command was introduced as of version 11.01.02.

2.15 GROUP SERVER RADIUS Configuration Menu Commands

You can configure the groups of Radius servers in the GROUP SERVER RADIUS configuration menu.

This menu can be accessed from the AAA feature configuration menu.

```
AAA config>group server radius GrupoRad
Radius group GrupoRad>
```

The following commands are available:

Command	Function
? (HELP)	Displays the configuration commands or their options.

NO	Negates a command or establishes its default parameters.
SERVER	Adds a server to a group of servers.
EXIT	Exits the configuration menu.

The following paragraphs describe each of the above commands.

2.15.1 ? (HELP)

Displays the available commands or their options.

2.15.2 NO

Configures parameters using the default values or deletes the configuration.

Syntax:

```
Radius group groupname>no server <serverid>
```

Example:

```
Radius group GrupoRad>no server mySrvRad1
Radius group GrupoRad>
```

In the example, the *mySrvRad1* server is eliminated from the group of servers.

2.15.3 SERVER

Adds a Radius server to the group of servers.

Syntax:

```
Radius group groupname>server <serverid>
```

<serverid>:	Radius server identifier.

The servers are used in the same order they were added. For instance, AAA petitions, regardless of their type, are first sent to the server that was added first.

Example:

```
Radius group GrupoRad>server mySrvRad1
Radius group GrupoRad>server mySrvRad2
Radius group GrupoRad>
```

In the example, two servers (*mySrvRad1* and *mySrvRad2*) are added to the group of servers. When this group is used in a method list, the *mySrvRad1* server shall be selected first and, only when it fails, shall the second server, *mySrvRad2,* be selected.

The servers must be defined beforehand. Please see the RADIUS-SERVERS configuration menu for further information.

2.16 GROUP SERVER TACACS+ Configuration Menu

The Tacacs+ server groups are configured in the GROUP SERVER TACACS+ configuration menu.

This configuration menu is accessed from the AAA feature configuration menu.

```
AAA config>group server tacacs+ GrupoTac
Tacacs+ group GrupoTac>
```

The following commands are available:

Command	**Function**
? (HELP)	Displays the configuration commands or their options.
NO	Negates a command or establishes its default parameters.
SERVER	Adds a server to a group of servers.
EXIT	Exits the configuration menu.

The following paragraphs describe each of the above commands.

2.16.1 ? (HELP)

Displays the available commands or their options.

2.16.2 NO

Configures parameters using the default values or deletes the configuration.

Syntax:

```
Tacacs+ group groupname>no server <serverid>
```

Example:

```
Tacacs+ group GrupoTac>no server mySrvTac1
Tacacs+ group GrupoTac>
```

In the example, the *mySrvTac1* server is eliminated from the group of servers.

2.16.3 SERVER

Adds a Tacacs+ server to the group of servers.

Syntax:

```
Tacacs+ group groupname>server <serverid>
```

<serverid>:	Tacacs+ server identifier

The servers are used in the same order in which they were added. For instance, AAA petitions, regardless of their type, are first sent to the server that was added first.

Example:

```
Tacacs+ group GrupoTac>server mySrvTac1
Tacacs+ group GrupoTac>server mySrvTac2
Tacacs+ group GrupoTac>
```

In the example, two servers (*mySrvTac1* and *mySrvTac2*) are added to the group of servers. When this group is used in a method list, the *mySrvTac1* server shall be selected first and, only when it fails, shall the second server, *mySrvTac2,* be used.

Servers must be defined beforehand. For further information, please see the section on the TACACS-SERVERS configuration menu.

2.17 RADIUS-SERVERS Configuration Menu

The Radius servers used in the AAA feature are configured in the RADIUS-SERVERS configuration menu.

The Radius server root configuration menu can be accessed from the AAA feature configuration menu.

```
AAA config>radius-servers
Radius servers>
```

The following commands are available:

Command	Function
? (HELP)	Displays the configuration commands or their options.
ATTRIBUTE	Allows extra attributes to be added to the Radius request.
KEY	Sets the key for the Radius servers, which take the default value in this field.
NO	Negates a command or establishes its default parameters.
PORT	Establishes the Radius servers' port taking its default value in this field.
SERVER	Accesses the configuration menu for a Radius server.
TIMEOUT	Establishes the wait time the Radius servers have as the default value in this field.
EXIT	Exits the configuration menu.

The purpose of these commands is to simplify and speed up the servers' configuration. The changes made in said menu affect all the Radius servers with default values. For instance, if you configure port 1234 in this menu, then all servers with a default port will use value 1234.

Note

The commands in the Radius servers' configuration root menu affect all Radius servers (unless they have a different value from the default one).

From this menu, you can access the configuration menu for an individual Radius server.

```
Radius servers>server mySrvRad1
Radius serv mySrvRad1>
```

The following commands are available:

Command	Function
? (HELP)	Displays the configuration commands or their options.
ATTRIBUTE	Allows extra attributes to be added to a Radius request.
HOST	Establishes the IP address for this server.
KEY	Establishes the key for this server.
NO	Negates a command or establishes its default parameters.
PORT	Establishes the port for this server.
SOURCE-ADDRESS	Establishes the source address to be used in the petitions to this server.
TIMEOUT	Establishes the time waited for this server.
EXIT	Exits the configuration menu.

The following paragraphs describe each of the above commands.

2.17.1 ? (HELP)

Displays the available commands or their options.

2.17.2 ATTRIBUTE

Allows extra attributes to be added to the Radius request.

Syntax:

```
Radius serv servername>attribute ?
  acct-authentic     Accounting-Requests include acct-authentic
  acct-session-id    Access-Requests include Accounting ID
  event-timestamp    Accounting-Requests include event-timestamp
  nas-id             All-Requests include configured NAS identification
```

Command history

Release	Modification
11.00.04	The "*attribute*" command was introduced as of version 11.00.04.
11.01.00	The "*attribute*" command was introduced as of version 11.01.00.
11.00.06	Command options **acct-authentic**, **acct-session-id** and **event-timestamp** have been added.
11.01.02	Command options **acct-authentic**, **acct-session-id** and **event-timestamp** have been added.

2.17.2.1 ATTRIBUTE ACCT-AUTHENTIC

Enables the sending of *Acct-Authentic* (attribute #45) in Radius Accounting-Request packets.

Syntax:

```
Radius serv servername>attribute acct-authentic
```

Command history

Release	Modification
11.00.06	The "*attribute acct-authentic*" command was introduced as of version 11.00.06.
11.01.02	The "*attribute acct-authentic*" command was introduced as of version 11.01.02.

2.17.2.2 ATTRIBUTE ACCT-SESSION-ID

Enables the sending of *Acct-Session-Id* (attribute #44) in Radius Access-Request packets.

Syntax:

```
Radius serv servername>attribute acct-session-id
```

Command history

Release	Modification
11.00.06	The "*attribute acct-session-id*" command was introduced as of version 11.00.06.
11.01.02	The "*attribute acct-session-id*" command was introduced as of version 11.01.02.

2.17.2.3 ATTRIBUTE EVENT-TIMESTAMP

Enables the sending of *Event-Timestamp* (attribute #55) in Radius Accounting-Request packets.

Syntax:

```
Radius serv servername>attribute event-timestamp
```

Command history

Release	Modification
11.00.06	The "*attribute event-timestamp*" command was introduced as of version 11.00.06.
11.01.02	The "*attribute event-timestamp*" command was introduced as of version 11.01.02.

2.17.2.4 ATTRIBUTE NAS-ID

Sets NAS-ID (attribute number 32) to a certain value, added to the Radius access request.

Syntax:

```
Radius serv servername>attribute nas-id <value>
```

Example:

```
Radius serv mySrvRad1>attribute nas-id mydomain
Radius serv mySrvRad1>
```

In this example, the attribute number NAS-ID is set to a value and is subsequently sent to the Radius server.

Command history

Release	Modification
11.00.04	The "*attribute nas-id*" command was introduced as of version 11.00.04.
11.01.00	The "*attribute nas-id*" command was introduced as of version 11.01.00.

2.17.3 HOST

Defines the IP address for the Radius server.

Syntax:

```
Radius serv servername>host <ipaddress> [vrf <vrf_name>]
```

<ipaddress>:	Server IP address.
<vrf_name>:	VRF used to communicate with the server.

Example:

```
Radius serv mySrvRad1>host 1.1.1.1
Radius serv mySrvRad1>
```

In this example, the server's IP address is defined as 1.1.1.1.

2.17.4 KEY

Defines the key to exchange traffic with this server.

Syntax:

```
Radius serv servername>key {plain | ciphered} <key>
```

plain:	This key is not encoded.
ciphered:	This key is encrypted.
<key>:	Server key.

The *plain* option specifies a plain key and the *ciphered* option specifies an encrypted key.

Example 1:

```
Radius serv mySrvRad1>key plain 1234
Radius serv mySrvRad1>
```

Example 1 sets a plain key.

Example 2:

```
Radius serv mySrvRad1>key ciphered 0x4698601DE5BFA77D
Radius serv mySrvRad1>
```

Example 2 sets an encrypted key.

Note

If the key takes the default value, it takes it from the root menu configuration located in the Radius Servers configuration.

2.17.5 NO

Configures parameters using the default values or deletes the configuration.

Syntax:

```
Radius serv servername>no host
Radius serv servername>no key
Radius serv servername>no port
Radius serv servername>no source-address
Radius serv servername>no timeout
```

Example:

```
Radius serv mySrvRad1>no timeout
```

In this example, the timeout takes its default value.

2.17.6 PORT

Establishes the Radius server port. Default is 1812.

Syntax:

```
Radius serv servername>port <port>
```

<port>:	Port.

Example:

```
Radius serv mySrvRad1>port 5000
Radius serv mySrvRad1>
```

This example sets the port to 5000.

Note

If the port takes the default value, it takes it from the root menu configuration located in the Radius Servers configuration.

2.17.7 SOURCE-ADDRESS

Establishes the IP address or the interface the router will use to communicate with the Radius server.

Syntax:

```
Radius serv servername>source-address {<ipaddress> | <interface>}
```

<ipaddress>:	IP address used to communicate with the server.
<Interface>:	Interface used to communicate with the server.

Example:

```
Radius serv mySrvRad1>source-address ethernet0/0
Radius serv mySrvRad1>
```

This example specifies that the router must communicate with the *mySrvRad1* server through its ethernet0/0 interface.

2.17.8 TIMEOUT

This command specifies the time (in seconds) the router must wait for a response from the Radius server. When using Radius, the UDP protocol makes three attempts before considering the server is down. Default is 5 seconds.

Syntax:

```
Radius serv servername>timeout <value>
```

<value>:	Wait time in seconds.

Example:

```
Radius serv mySrvRad1>timeout 10
Radius serv mySrvRad1>
```

This example sets the timeout to 10 seconds.

Note

If the timeout takes the default value, it takes it from the root menu configuration located in the Radius Servers configuration.

2.18 TACACS-SERVERS Configuration Menu Commands

In the TACACS-SERVERS configuration menu, you configure the Tacacs+ servers that are going to be used in the AAA feature.

You can access the AAA feature configuration menu through the Tacacs+ servers' root menu configuration.

```
AAA config>tacacs-servers
Tacacs+ servers>
```

The following commands are available:

Command	Function
? (HELP)	Displays the configuration commands or their options.

KEY	Sets the Tacacs+ servers' key, taking its default value in this field.
NO	Negates a command or establishes its default parameters.
PORT	Sets the Tacacs+ servers' port, taking its default value in this field.
SERVER	Accesses the configuration menu for a Tacacs+ server.
TIMEOUT	Sets the Tacacs+ servers' timeout, taking its default value in this field.
USERNAME-SUFFIX	Sets the Tacacs+ servers' suffix that must be added to the username.
EXIT	Exits the configuration menu.

These commands aim at simplifying and speeding up the servers' configuration. The changes made in this menu affect all Tacacs+ servers with default values. For example, if you configure port 4321 in this menu, then all servers with a default port will use value 4321.

Note

The commands in the Tacacs+ servers' configuration root menu affect all Tacacs+ servers (unless they have a different value from the default one).

From this menu, you can access the configuration menu for an individual Tacacs+ server.

```
Tacacs+ servers>server mySrvTac1
Tacacs+ serv mySrvTac1>
```

The following commands are available:

Command	Function
? (HELP)	Displays the configuration commands or their options.
HOST	Establishes the IP address for this server.
KEY	Establishes the key for this server.
NO	Negates a command or establishes its default parameters.
PORT	Establishes the port for this server.
SOURCE-ADDRESS	Establishes the source address to use in the petitions to this server.
TIMEOUT	Establishes the time waited for this server.
USERNAME-SUFFIX	Establishes the Tacacs+ servers' suffix to add to the username.
EXIT	Exits the configuration menu.

The following paragraphs describe each of the above commands.

2.18.1 ? (HELP)

Displays the available commands or their options.

2.18.2 HOST

Defines the IP address for the Tacacs+ server.

Syntax:

```
Tacacs+ serv servername>host <ipaddress> [vrf <vrf_name>]
```

<ipaddress>:	Server IP address.
<vrf_name>:	VRF used to communicate with the server.

Example:

```
Tacacs+ serv mySrvTac1>host 2.2.2.2
Tacacs+ serv mySrvTac1>
```

This example defines the server's IP address as 2.2.2.2.

2.18.3 KEY

Defines the key to exchange traffic with this server.

Syntax:

```
Tacacs+ serv servername>key {plain | ciphered} <key>
```

plain:	This key is not encoded.
ciphered:	The key is encrypted.
<key>:	Server key.

The *plain* option specifies the plain key and the *ciphered* option specifies the encrypted key.

Example 1:

```
Tacacs+ serv mySrvTac1>key plain abcd
Tacacs+ serv mySrvTac1>
```

Example 1 sets the plain key.

Example 2:

```
Tacacs+ serv mySrvTac1>key ciphered 0x413148BFDA8F9860
Tacacs+ serv mySrvTac1>
```

Example 2 sets the encrypted key.

Note

If the key takes its default value, it takes the configuration from the root menu of the Tacacs+ servers' configuration.

2.18.4 NO

Configures parameters using the default values or deletes the configuration.

Syntax:

```
Tacacs+ serv servername>no host
Tacacs+ serv servername>no key
Tacacs+ serv servername>no port
Tacacs+ serv servername>no source-address
Tacacs+ serv servername>no timeout
```

Example:

```
Tacacs+ serv mySrvTac1>no source-address
Tacacs+ serv mySrvTac1>
```

This example establishes the source address for petitions to the server as default.

2.18.5 PORT

Establishes the Tacacs+ server port. Default is 49.

Syntax:

```
Tacacs+ serv servername>port <port>
```

<port>:	Port.

Example:

```
Tacacs+ serv mySrvTac1>port 6000
Tacacs+ serv mySrvTac1>
```

This example sets the port to value 6000.

Note

If the port takes its default value, it takes the configuration from the root menu of the Tacacs+ servers' configuration.

2.18.6 SOURCE-ADDRESS

Establishes the IP address or interface used by the router to communicate with the Tacacs+ server.

Syntax:

```
Tacacs+ serv servername>source-address {<ipaddress> | <interface>}
```

<ipaddress>:	IP address used to communicate with the server.
<Interface>:	Interface used to communicate with the server.

Example:

```
Tacacs+ serv mySrvTac1>source-address ethernet0/0
Tacacs+ serv mySrvTac1>
```

In this example, the router must communicate with the *mySrvTac1* server through its ethernet0/0 interface.

2.18.7 TIMEOUT

Sets the time (in seconds) the router waits for a response from the Tacacs+ server before considering that the server won't respond. Default is 5 seconds.

Syntax:

```
Tacacs+ serv servername>timeout <value>
```

<value>:	Wait time in seconds.

Example:

```
Tacacs+ serv mySrvTac1>timeout 15
Tacacs+ serv mySrvTac1>
```

This example sets the timeout to 15 seconds.

Note

If the timeout takes its default value, it takes the configuration from the root menu of the Tacacs+ servers' configuration.

2.18.8 USERNAME-SUFFIX

Sets the suffix that must be added to the username before sending it to the Tacacs+ server.

Syntax:

```
Tacacs+ serv servername>username-suffix {plain | ciphered | ciphered-unique} <suffix>
```

plain:	This suffix is not encoded.
ciphered:	This suffix is ciphered.
ciphered-unique:	This suffix is ciphered in a unique way for this device.
<suffix>:	Suffix to be added

Example:

```
Tacacs+ serv mySrvTac1>username-suffix plain @domain.com
Tacacs+ serv mySrvTac1>
```

This example sets the suffix to "@domain.com".

Note

If the username-suffix takes its default value, it takes the configuration from the root menu of the Tacacs+ servers' configuration.

Command history:

Release	Modification
10.8.34.5.12	The "*username-suffix*" command was introduced as of version 10.8.34.5.12.
11.00.05	The "*username-suffix*" command was introduced as of version 11.00.05.
11.01.01	The "*username-suffix*" command was introduced as of version 11.01.01.

2.19 Using the AAA feature in router services

A series of commands relative to the AAA feature may be found in the configuration menu for each service. The purpose of these commands is to apply the method lists to several services so that they can use the features offered by AAA.

2.19.1 Using the AAA feature in the console

This section focuses on how console configuration commands linked to the AAA feature are used. For further information on this service, please see manual *bintec-Dm704-I Configuration Monitoring*.

Access the console configuration menu through the router's main configuration menu.

Syntax:

```
Config>set console
```

Example:

```
Config>set console
-- Console configuration --
Con config>
```

The console service supports various method lists. The commands are as follows:

Command	Function
ACCOUNTING	Establishes an *accounting* method list.
AUTHORIZATION	Establishes an *authorization* method list.
LOGIN AUTHENTICATION	Establishes an *authentication* method list.

The following paragraphs describe each of the above commands.

2.19.1.1 ACCOUNTING

Associates an *accounting exec* or *commands* method list to the console service. Consequently, the console service applies the *accounting exec* method list when access to the console shell is registered, and the *accounting commands* method when a command is executed from the console.

Syntax:

```
Con config>accounting {commands <level> | exec} <listname>
```

commands:	Associates an *accounting commands* method list.
<level>:	Access level for the commands that require *accounting*.
exec:	Associates an *accounting exec* method list.
<listname>:	Identifier for the *accounting* method list.

Example 1:

```
Con config>accounting commands 10 AccCmds
Con config>
```

In example 1, the *AccCmds* method list is configured so that, when a level 10 command is executed in the console and *accounting* is carried out, the list is used.

Example 2:

```
Con config>accounting exec AccExec
Con config>
```

In example 2, the *AccExec* method list is configured so that, when *accounting* is carried out after the console shell is accessed, the aforementioned list is used.

2.19.1.2 AUTHORIZATION

Associates an *authorization exec* or *commands* method list to the console service. Consequently, the console service applies the *authorization exec* method list when *authorization* is required from the shell, and *authorization commands* when *authorization* is required from a command.

Syntax:

```
Con config>authorization {commands <level> | exec} <listname>
```

commands:	Associates an *accounting commands* method list.
<level>:	Access level for the commands that require *accounting*.
exec:	Associates an *accounting exec* method list.
<listname>:	Identifier for the *accounting* method list.

Example 1:

```
Con config>authorization commands 10 AuthorCmds
Con config>
```

In example 1, the *AuthorCmds* method list is configured so that it is used when *authorization* is required for level 10 commands in the console.

Example 2:

```
Con config> authorization exec AuthorExec
Con config>
```

In example 2, the *AuthorExec* method list is configured to be used when *authorization* is required from the console shell.

2.19.1.3 LOGIN AUTHENTICATION

Associates an *authentication login* method list to the console service. Consequently, the console service applies the associated method list when authentication needs to be carried out.

Syntax:

```
Con config>login authentication <listname>
```

<listname>:	Identifier for the *authentication* method list.

Example:

```
Con config>login authentication AutheLogin
Con config>
```

In the example, the *AutheLogin* method list is configured to be used when authentication is required for console-based user access.

2.19.2 Using the AAA feature in Telnet

This section focuses on how Telnet configuration commands linked to the AAA feature are used. For more information about this service, please see manual *bintec-Dm738-I TELNET Protocol*.

The Telnet configuration menu can be accessed through the router's main configuration menu.

Syntax:

```
Config>set telnet
```

Example:

```
Config>set telnet
-- Telnet user configuration --
Telnet config>
```

The Telnet service supports various method lists. The commands are as follows:

Command	Function
ACCOUNTING	Establishes an *accounting* method list.
AUTHORIZATION	Establishes an *authorization* method list.
LOGIN AUTHENTICATION	Establishes an *authentication* method list.

The following paragraphs describe each of the above commands.

2.19.2.1 ACCOUNTING

Associates an *accounting exec* or *commands* method list to the Telnet service. Consequently, the Telnet service applies the *accounting exec* method list when registering an access to the Telnet shell, and the *accounting commands* method when a command is executed through Telnet.

Syntax:

```
Telnet config>accounting {commands <level> | exec} <listname>
```

commands:	Associates an *accounting commands* method list.
<level>:	Access level for the commands that require *accounting*.
exec:	Associates an *accounting exec* method list.
<listname>:	Identifier for the *accounting* method list.

Example 1:

```
Telnet config>accounting commands 15 AccCmds
Telnet config>
```

In example 1, the *AccCmds* method list is configured so that, when *accounting* is executed for a level 15 command executed from Telnet, the aforementioned list is used.

Example 2:

```
Telnet config>accounting exec AccExec
Telnet config>
```

In example 2, the *AccExec* method list is configured so that, when the Telnet shell is accessed and *accounting* is carried out, the aforementioned list is used.

2.19.2.2 AUTHORIZATION

Associates an *authorization exec* or *commands* method list to the Telnet service. Consequently, the Telnet service applies the *authorization exec* method list when authorization is required by the shell, and *authorization commands* when authorization is required by a command.

Syntax:

```
Telnet config>authorization {commands <level> | exec} <listname>
```

commands:	Associates an *accounting commands* method list.
<level>:	Access level for the commands that require *accounting*.
exec:	Associates an *accounting exec* method list.
<listname>:	Identifier for the *accounting* method list.

Example 1:

```
Telnet config>authorization commands 15 AuthorCmds
Telnet config>
```

In example 1, the *AuthorCmds* method list is configured to be used when authorization is required for level 15 commands in Telnet.

Example 2:

```
Telnet config>authorization exec AuthorExec
Telnet config>
```

In example 2, the *AuthorExec* method list is configured to be used when authorization is required by the Telnet shell.

2.19.2.3 LOGIN AUTHENTICATION

Associates an *authentication login* method list to the Telnet service. Consequently, the Telnet service applies the associated method list when authentication needs to be carried out.

Syntax:

```
Telnet config>login authentication <listname>
```

<listname>:	Identifier for the *authentication* method list.

Example:

```
Telnet config>login authentication AutheLogin
Telnet config>
```

In this example, an *AutheLogin* method list is configured to be used when authentication is required for a user accessing through Telnet.

2.19.3 Using the AAA feature in FTP

This section focuses on how FTP configuration commands linked to the AAA feature are used. For more information about this service, please see manual *bintec-Dm724-I FTP Protocol.*

The FTP configuration menu can be accessed through the router's main configuration menu.

Syntax:

```
Config>set ftp
```

Example:

```
Config>set ftp
-- FTP user configuration --
FTP config>
```

The FTP service supports various method lists. The commands are as follows:

Command	Function
ACCOUNTING	Establishes an *accounting* method list.
AUTHORIZATION	Establishes an *authorization* method list.
LOGIN AUTHENTICATION	Establishes an *authentication* method list.

The following paragraphs describe each of the above commands.

2.19.3.1 ACCOUNTING

Associates an *accounting exec* method list to the FTP service. Consequently, the FTP service applies the *accounting exec* method list when an access to the FTP shell is registered.

Syntax:

```
FTP config>accounting exec <listname>
```

exec:	Associates an *accounting exec* method list.
<listname>:	Identifier for the *accounting* method list.

Example:

```
FTP config>accounting exec AccExec
FTP config>
```

In the example, the *AccExec* method list is configured so that, when *accounting* is carried out after having accessed the FTP shell, the aforementioned list is used.

2.19.3.2 AUTHORIZATION

Associates an *authorization exec* method list to the FTP service. Consequently, the FTP service applies the *authorization exec* method list when the shell requires authorization.

Syntax:

```
FTP config>authorization exec <listname>
```

exec:	Associates an *accounting exec* method list.
<listname>:	Identifier for the *authorization* method list.

Example:

```
FTP config>authorization exec AuthorExec
FTP config>
```

In the example, the *AuthorExec* method list is configured to be used when the FTP shell requires authorization.

2.19.3.3 LOGIN AUTHENTICATION

Associates an *authentication login* method list to the FTP service. Consequently, the FTP service applies the associated method list when authentication needs to be carried out.

Syntax:

```
FTP config>login authentication <listname>
```

<listname>:	Identifier for the *authentication* method list.

Example:

```
FTP config>login authentication AutheLogin
FTP config>
```

In the example, the *AutheLogin* method list is configured to be used when authentication is required for a user accessing through FTP.

2.19.4 Using the AAA feature in SSH

This section focuses on how SSH configuration commands linked to the AAA feature are used. For more information about this service, please see manual *bintec-Dm787-I SSH Protocol.*

The SSH server configuration menu can be accessed through the router's main configuration menu.

Syntax:

```
Config>feature ssh
SSH Config>server
```

Example:

```
Config>feature ssh
-- SSH protocol configuration --
SSH Config>server
-- SSH Server --
SSHS>
```

The SSH service supports various method lists. The commands are as follows:

Command	Function
ACCOUNTING	Establishes an *accounting* method list.
AUTHORIZATION	Establishes an *authorization* method list.
LOGIN AUTHENTICATION	Establishes an *authentication* method list.

Note

SSH only applies the *authentication* and/or *authorization* method lists when authenticating/authorizing clients by means of a password or, as of version 11.01.10, a public key.

The following paragraphs describe each of the above commands.

2.19.4.1 ACCOUNTING

Associates an *accounting exec* or *commands* method list to the SSH service. Consequently, the SSH service applies the *accounting exec* method list when access to the SSH shell is registered, and the *accounting commands* method when a command is executed from SSH.

Syntax:

```
SSHS config>accounting {commands <level> | exec} <listname>
```

commands:	Associates an *accounting commands* method list.
<level>:	Access level for the commands that require *accounting*.
exec:	Associates an *accounting exec* method list.
<listname>:	Identifier for the *accounting* method list.

Example 1:

```
SSHS config>accounting commands 5 AccCmds
SSHS config>
```

In example 1, the *AccCmds* method list is configured so that, when a level 5 command is executed from SSH and *accounting* is carried out, the aforementioned list is used.

Example 2:

```
SSHS config>accounting exec AccExec
SSHS config>
```

In example 2, the *AccExec* method list is configured so that, when the SSH shell is accessed and *accounting* is carried out, the aforementioned list is used.

2.19.4.2 AUTHORIZATION

Associates an *authorization exec* or *commands* method list to the SSH service. Consequently, the SSH service applies the *authorization exec* method list when the shell requires authorization, and *authorization commands* when a command requires authorization.

Syntax:

```
SSHS config>authorization {commands <level> | exec} <listname>
```

commands:	Associates an *authorization commands* method list.
<level>:	Access level for the commands that require authorization.
exec:	Associates an *authorization exec* method list.
<listname>:	Identifier for the *authorization* method list.

Example 1:

```
SSHS config>authorization commands 10 AuthorCmds
SSHS config>
```

In example 1, the *AuthorCmds* method list is configured to be used when level 10 commands in SSH require authorization.

Example 2:

```
SSHS config>authorization exec AuthorExec
SSHS config>
```

In example 2, the *AuthorExec* method list is configured to be used when the SSH shell requires authorization.

2.19.4.3 LOGIN AUTHENTICATION

Associates an *authentication login* method list to the SSH service. The SSH service then applies the associated method list when authentication needs to be carried out.

Syntax:

```
SSHS config>login authentication <listname>
```

<listname>:	Identifier for the *authentication* method list.

Example:

```
SSHS config>login authentication AutheLogin
SSHS config>
```

In the example, the *AutheLogin* method list is configured to be used when authentication is required for users accessing through SSH.

2.19.5 Using the AAA feature in PPP links

This section focuses on how PPP links configuration commands linked to the AAA feature are used. For further information, please see manual *Dm710-I PPP Interface*.

With PPP encapsulation enabled, the interfaces can negotiate with a client to establish a PPP session, which provides access to network services. To execute said negotiation and allow the client to access his specific services, PPP needs to authenticate the client's connection. For further information on how to configure PPP in the router, please see manual *bintecDm710-I PPP Interface.*

Access the PPP interface configuration menu through the router's main configuration menu.

Syntax:

```
Config>network <pppN>
```

<pppN>:	PPP interface identifier for the router.

Example:

```
Config>network ppp1
-- Generic PPP User Configuration –
ppp1 config>
```

The AAA configuration commands can be found in the submenu, which is accessed through the *ppp* command.

Syntax:

```
<pppN> config>ppp
```

Example:

```
ppp1 config>ppp
-- PPP Configuration --
ppp1 PPP config>
```

PPP links support various method lists. The commands are as follows:

Command	Function
AUTHENTICATION {CHAP | PAP}	Establishes an *accounting* method list.
AUTHORIZATION	Establishes an *authorization* method list.

2.19.5.1 AUTHENTICATION {CHAP | PAP}

Configures the type of authentication for PPP links. It runs the *authentication ppp* method list so that it is used, together with the CHAP and PAP protocols, in connection authentication.

Syntax:

```
<pppN> PPP config>authentication {chap | pap} <listname>
```

<listname>:	Identifier for the *authentication* method list.

Example:

```
ppp1 PPP config>authentication chap AuthePPP
ppp1 PPP config>
```

In this example, the *AuthePPP* list is applied to the PPP interface to be used when authenticating connections.

2.19.5.2 AUTHORIZATION

Associates an *authorization network* method list to the interface. This way, the PPP link applies the method list when it needs authorizing.

Syntax:

```
<pppN> PPP config>authorization network <listname>
```

<listname>:	Identifier for the *authorization* method list.

Example:

```
ppp1 PPP config>authorization network AuthorNet
ppp1 PPP config>
```

In this example, the *AuthorNet* method list is configured to be used when the link requires authorization.

2.19.6 Using the AAA feature in the HotSpot feature

This section focuses on how to use the AAA feature in the HotSpot configuration commands. For more information about this feature, please see manual *bintec-Dm820-I Hotspot Feature*.

The HotSpot feature grants network access to physical routers, also called subscribers, based on their user credentials. To do this, a session, which is associated with a subscriber's MAC address, is maintained for each subscriber and given certain privileges based on the credentials submitted. The services offered by the AAA feature are used to validate user credentials, set privileges and carry out statistical *accounting* for each session. For further information, please see manual *bintec-Dm820-I Hotspot Feature*.

Access the HotSpot feature configuration menu from the router's main configuration menu.

Syntax:

```
Config>feature hotspot
```

Example:

```
Config>feature hotspot
-- Hotspot Configuration -
Hs config>
```

The AAA configuration commands can be found, in certain network interfaces, in the HotSpot feature configuration submenu. You may access the latter through the *network <interfaz>* command.

Syntax:

```
HS config>network <Interfaz>
```

Example:

```
HS config>network wlan0/0
Network wlan0/0>
```

The HotSpot facility supports various method lists. The following commands can be used for each of them:

Command	Function
ACCOUNTING	Establishes an *accounting* method list.
AUTHENTICATION	Establishes an *authentication* method list.
AUTHORIZATION	Establishes an *authorization* method list.

2.19.6.1 ACCOUNTING

Associates an *accounting network* method list with the HotSpot service. When it needs *accounting* statistics for a subscriber session, the HotSpot service applies the methods on the *accounting network* list.

Syntax:

```
Network <Interface>>accounting network <listname>
```

<listname>:	Name of the *accounting* method list.

Example:

```
Network wlan0/0>accounting network AcctNet
Network wlan0/0>
```

In the example, the *AcctNet* method list is configured to be used when *accounting* statistics for a subscriber session are required.

Command history:

Release	Modification
11.00.03	This command was introduced as of version 11.00.03.

2.19.6.2 AUTHENTICATION

Associates an *authentication login* method list with the HotSpot service. When it needs to carry out authentication, the HotSpot service applies the methods on the list taking into account the user credentials submitted by a subscriber.

Syntax:

```
Network <Interface>>authentication login <listname>
```

<listname>:	Name of the *authentication* method list.

Example:

```
Network wlan0/0>authentication login AuthList
Network wlan0/0>
```

In the example, the *AuthList* is applied to the HotSpot service enabled on the wlan0/0 interface. This is used to authenticate subscriber sessions.

Command history:

Release	Modification
11.00.03	This command was introduced as of version 11.00.03.

2.19.6.3 AUTHORIZATION

Associates an *authorization network* method list with the HotSpot service.When it needs authorization for a subscriber session, the HotSpot service applies the methods on the *authorization network* list.

Syntax:

```
Network <Interface>>authorization network <listname>
```

<listname>:	Name of the *authorization* method list.

Example:

```
Network wlan0/0>authorization network AuthorNet
Network wlan0/0>
```

In the example, the *AuthorNet* method list is configured to be used when a subscriber session requires authorization.

Command history:

Release	Modification
11.00.03	This command was introduced as of version 11.00.03.

2.19.7 Using the AAA feature in HTTP

This section focuses on how the HTTP configuration commands are used in connection with the AAA feature. For more information about this service, please see manual *bintec-Dm737-I HTTP Protocol.*

The HTTP configuration menu can be accessed through the router's main configuration menu.

Syntax:

```
Config>feature http
```

Example:

```
Config>feature http

-- HTTP user configuration --
HTTP config>
```

The HTTP service supports several method lists. The commands are as follows:

Command	Function
AUTHORIZATION	Establishes an *authorization* method list.
LOGIN AUTHENTICATION	Establishes an *authentication* method list.

The following sections describe each of the above commands.

2.19.7.1 AUTHORIZATION

Associates an *authorization exec* method list to the HTTP service. This way, the HTTP service applies the method list for authorization purposes.

Syntax:

```
HTTP config>authorization exec <listname>
```

<listname>:	Identifier for the *authorization* method list.

Example:

```
HTTP config>authorization exec AuthorExec
HTTP config>
```

In the example, the *AuthorExec* method list is configured to be used by the HTTP service for authorization purposes.

Command history:

Release	Modification
10.08.34.05.12	This command was introduced as of version 10.08.34.05.12.
11.00.05	This command was introduced as of version 11.00.05.
11.01.01	This command was introduced as of version 11.01.01.

2.19.7.2 LOGIN AUTHENTICATION

Associates an *authentication login* method list to the HTTP service. Thus, when authentication needs to be carried out, the HTTP service applies the associated method list.

Syntax:

```
HTTP config>login authentication <listname>
```

<listname>:	Identifier for the *authentication* method list.

Example:

```
HTTP config>login authentication AuthLogin
HTTP config>
```

In this example, an *AuthLogin* method list is configured to be used when authentication is required for users accessing through HTTP.

Command history:

Release	Modification
10.08.34.05.09	This command was introduced as of version 10.08.34.05.09.
11.00.05	This command was introduced as of version 11.00.05.
11.01.00	This command was introduced as of version 11.01.00.

Chapter 3 Monitoring

Access the AAA feature monitoring menu through the main monitoring menu (PROCESS 3) by executing the *feature aaa* command.

Syntax:

```
+feature aaa
-- AAA User Console --
AAA +?
  list    List AAA information
  exit    Exit to parent menu
```

The following sections detail each command and what they are used for.

3.1 LIST

Displays information and statistics on the AAA feature.

Syntax:

```
AAA +list ?
  coa    List CoA information in AAA feature
```

Command history

Release	Modification
11.00.06	The "*list*" command was introduced as of version 11.00.06.
11.01.02	The "*list*" command was introduced as of version 11.01.02.

3.1.1 LIST COA

Displays information and statistics on the Change of Authorization (CoA) functionality, configured in the AAA feature.

Syntax:

```
AAA +list coa statistics
```

Example:

```
AAA +list coa statistics

==============================
..::  AAA CoA statistics  ::..
==============================
Rx msg 'Unknown': 0
Rx msg 'Disconnect': 0
Rx msg 'HotSpot:Reauthenticate': 0
Err 'No Error': 0
Err 'Residual Session Context Removed': 0
Err 'Invalid EAP Packet': 0
Err 'Unsupported Attribute': 0
Err 'Missing Attribute': 0
Err 'NAS Identification Mismatch': 0
Err 'Invalid Request': 0
Err 'Unsupported Service': 0
Err 'Unsupported Extension': 0
Err 'Invalid Attribute Value': 0
Err 'Administratively Prohibited': 0
Err 'Request Not Routable': 0
Err 'Session Context Not Found': 0
Err 'Session Context Not Removable': 0
Err 'Other Proxy Processing Error': 0
Err 'Resources Unavailable': 0
Err 'Request Initiated': 0
Err 'Multiple Session Selection Unsupported': 0
```

```
====================================
..::  AAA Radius CoA statistics  ::..
====================================
Rx pkts: 0
Err 'No error': 0
Err 'Invalid message length': 0
Err 'Invalid message length in RADIUS header': 0
Err 'Parsing RAIDUS packet failed': 0
Err 'Unknown RADIUS client': 0
Err 'Shared secret is incorrect': 0
Err 'Unknown RADIUS message code': 0
Err 'Response could not be sent': 0
Err 'Message-Authenticator could not be calculated': 0
Err 'Too long response length': 0
```

Command history

Release	Modification
11.00.06	The "*list coa*" command was introduced as of version 11.00.06.
11.01.02	The "*list coa*" command was introduced as of version 11.01.02.

Chapter 4 Configuration Examples

4.1 Tacacs+ Authentication for the Telnet Shell

Imagine a one-router scenario with the following requirements:

- Shell access is only possible, for previously authenticated users, through the router and through Telnet.
- The user database is located in a Tacacs+ server.
- If the Tacacs+ server access drops, check the router's internal user database.

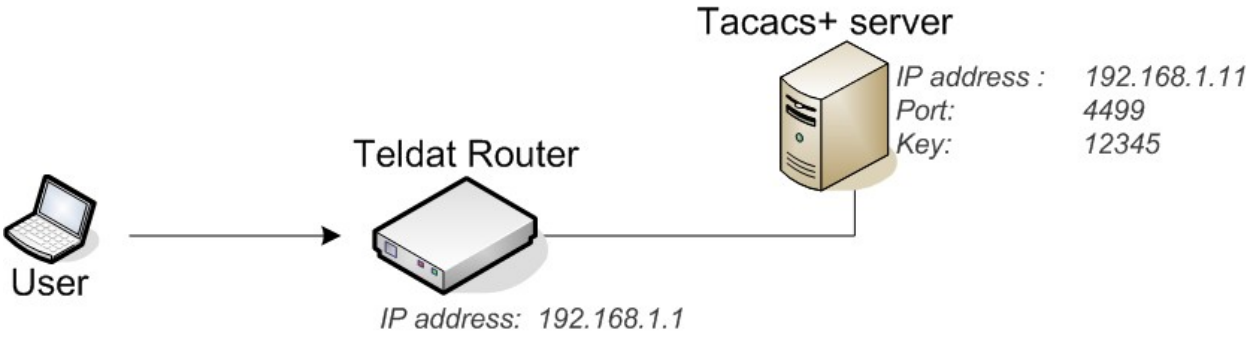

Fig. 7: **Scenario for Example 1.**

4.1.1 Creating the Tacacs+ Server

Firstly, register the Tacacs+ server in the AAA feature. To do this, access the AAA Tacacs+ servers menu:

```
Config>feature aaa
-- AAA user configuration --
AAA config>tacacs-servers
Tacacs+ servers>
```

Once in the menu, assign an identifier to the server (in this case, *T1*).

```
Tacacs+ servers>server T1
Tacacs+ serv T1>
```

Now, fill out the server parameters: IP address, port and key.

```
Tacacs+ serv T1>host 192.168.1.11
Tacacs+ serv T1>port 4499
Tacacs+ serv T1>key plain 12345
Tacacs+ serv T1>
```

4.1.2 Creating a group of servers

At this stage, it is time to create a group of Tacacs+ servers where server T1 is added. To do this, set a group identifier and access the menu. In this case, the group identifier is *GrupoTac1.*

```
AAA config>group server tacacs+ GrupoTac1
Tacacs+ group GrupoTac1>
```

Add the T1 server to this group.

```
Tacacs+ group GrupoTac1>server T1
Tacacs+ group GrupoTac1>
```

4.1.3 Creating local users

Register a user in the local database: user *root*, password *root1234*.

```
Config>user root password root1234
Config>
```

4.1.4 Creating a method list

Method lists that apply shell authentication are *authentication login*. They are created using the *AuthenLogin* identifier.

```
AAA config>authentication login AuthenLogin
Login list AuthenL...>
```

A maximum priority method is set, together with a *group* type (which refers to the recently created group of servers).

```
Login list AuthenL...>method 1 group GrupoTac1
Login list AuthenL...>
```

A priority 2 method is created, as well as a *local* type, to be applied to the local user database in case the first method fails.

```
Login list AuthenL...>method 2 local
Login list AuthenL...>
```

At this point, we already have a method list for shell authentication. There are two methods: the first refers to a Tacacs+ server and the second to the router's own database.

4.1.5 Enabling AAA

Before associating the method list to any service, enable the AAA feature.

```
AAA config>enable
AAA config>
```

4.1.6 Associating the method list to Telnet

The authentication restriction must only be applied to the Telnet service. Access said menu.

```
Config>set telnet
-- Telnet user configuration --
Telnet config>
```

Authentication is configured with the method list previously created.

```
Telnet config>login authentication AuthenLogin
Telnet config>
```

Configuration is now complete. Now, when a user accesses the router shell through Telnet, he undergoes an authentication process. A Tacacs+ server is consulted first and, if the server is down, the router database is responsible for verifying said authentication.

The next console shows the full configuration:

```
  log-command-errors
   no configuration
   set data-link x25 serial0/1
;
   feature aaa
; -- AAA user configuration --
      enable
      tacacs-servers
         server "T1"
            port 4499
            key ciphered 0xE13697A11E572446
            host 192.168.1.11
         exit
;
      exit
;
      group server tacacs+ "GrupoTac1"
         server T1
      exit
;
      authentication login "AuthenLogin"
         method 1 group GrupoTac1
```

```
        method 2 local
      exit
;
   exit
;
   network ethernet0/0
; -- Ethernet Interface User Configuration --
      ip address 192.168.1.1 255.255.255.0
;
   exit
;
   set telnet
; -- Telnet user configuration --
      login authentication AuthenLogin
   exit
;
   dump-command-errors
   end
```

4.2 Tacacs+ Authorization Commands for all Services that support this

4.2.1 Tacacs+ Authentication for the Telnet Shell

Imagine a one-router scenario with the following requirements:

- You need to control the level 10 commands execution (permit or deny).
- You can find these commands (permit or deny) by consulting a Tacacs+ server (T1). If this isn't possible, consult the backup Tacacs+ server (T2).
- If none of the servers respond, level 10 commands can be executed without restrictions.

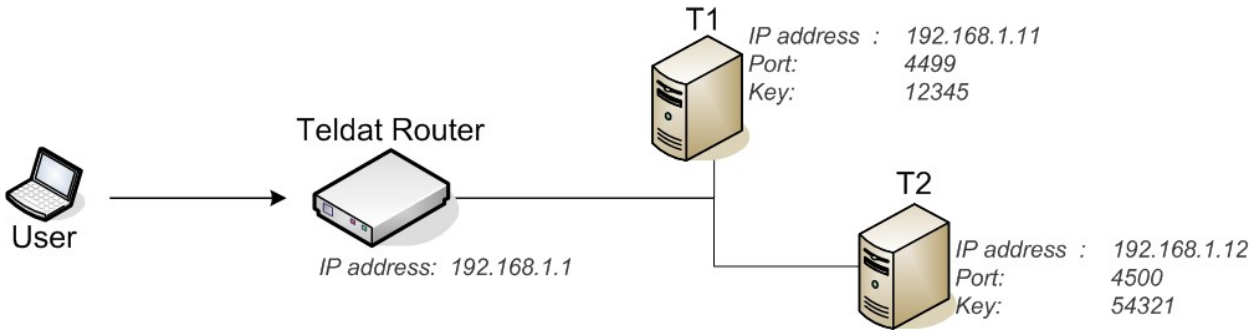

Fig. 8: **Scenario for Example 2.**

4.2.2 Creating the Tacacs+ Servers

Firstly, register the Tacacs+ servers in the AAA feature.

```
Config>feature aaa
-- AAA user configuration --
AAA config>tacacs-servers
Tacacs+ servers>server T1
Tacacs+ serv T1>host 192.168.1.11
Tacacs+ serv T1>port 4499
Tacacs+ serv T1>key plain 12345
Tacacs+ serv T1>exit
Tacacs+ servers>server T2
Tacacs+ serv T2>host 192.168.1.12
Tacacs+ serv T2>port 4500
Tacacs+ serv T2>key plain 54321
```

4.2.3 Creating the Servers Group

A group of Tacacs+ servers is created, having added the main server (T1) first and a backup server (T2) later. This group's identifier is *GrupoTac* .

```
AAA config>group server tacacs+ GrupoTac
Tacacs+ group GrupoTac>server T1
Tacacs+ group GrupoTac>server T2
Tacacs+ group GrupoTac>
```

4.2.4 Creating the method list

Method lists that run authorization commands are known as *authorization commands*. The method list you want to create must have the *default* identifier for it to be automatically applied to all services.

```
AAA config>authorization commands default
Cmds list default>
```

This accesses privilege level 10.

```
Cmds list default>privilege-level 10
Cmds lvl 10>
```

The recently created group is associated to the highest priority method and a *none* method is added. This way, no restrictions apply in situations where servers belonging to a group are down.

```
Cmds lvl 10>method 1 group GrupoTac
Cmds lvl 10>method 2 none
Cmds lvl 10>
```

4.2.5 Enabling AAA

The final step is to enable the AAA feature.

```
AAA config>enable
AAA config>
```

The router configuration is complete. Now forward a petition to the Tacacs+ server to authorize (or not) the execution of a privilege level 10 command.

The next console shows the full configuration:

```
  log-command-errors
   no configuration
   set data-link x25 serial0/1
;
   feature aaa
; -- AAA user configuration --
      enable
      tacacs-servers
         server "T1"
            port 4499
            key ciphered 0xE13697A11E572446
            host 192.168.1.11
         exit
;
         server "T2"
            port 4500
            key ciphered 0x660F5E0D4DD8714C
            host 192.168.1.12
         exit
;
      exit
;
      group server tacacs+ "GrupoTac"
         server T1
         server T2
      exit
;
```

```
      authorization commands "default"
         privilege-level 10
            method 1 group GrupoTac
            method 2 none
         exit
;
      exit
;
   exit
;
   network ethernet0/0
; -- Ethernet Interface User Configuration --
      ip address 192.168.1.1 255.255.255.0
;
   exit
;
   dump-command-errors
   end
```

4.3 Radius Authentication and Tacacs+ Accounting in the console

Imagine a one-router scenario with the following requirements:

- Shell access is only available, through the console, to previously authenticated users.
- The user database is located in a Radius server (R1).
- The privilege level 15 commands, executed in the console, must be registered in a Tacacs+ server (T1).

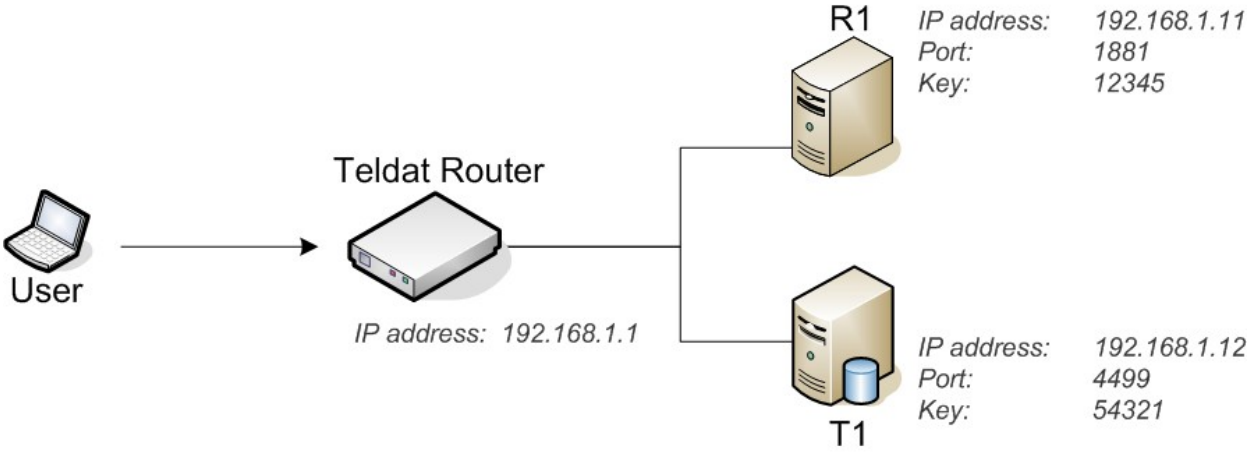

Fig. 9: **Scenario for Example 3.**

4.3.1 Creating the Radius Server

Firstly, register the Radius server in the AAA feature. To do this, access the AAA Radius servers menu.

```
Config>feature aaa
-- AAA user configuration --
AAA config>radius-servers
Radius servers>
```

Once in the menu, assign an identifier to the server (in this case, *R1*).

```
Radius servers>server R1
Radius serv R1>
```

Now fill out the server parameters: IP address, port and key.

```
Radius serv R1>host 192.168.1.11
Radius serv R1>port 1881
Radius serv R1>key plain 12345
Radius serv R1>
```

4.3.2 Creating the Tacacs+ Server

You also need to register the Tacacs+ server. This is very similar to registering the Radius server. The identifier in this case is *T1*.

```
AAA config>tacacs-servers
Tacacs+ servers>server T1
Tacacs+ serv T1>host 192.168.1.12
Tacacs+ serv T1>port 4499
Tacacs+ serv T1>key plain 54321
```

4.3.3 Creating the group of Radius Servers

At this stage, it is time to create a group of Radius servers where server R1 is added. To do this, set a group identifier and access the menu. In this case, the group identifier is *GrupoRad*.

```
AAA config>group server radius GrupoRad
Radius group GrupoRad>
```

The R1 server is added to the group.

```
Radius group GrupoRad>server R1
Radius group GrupoRad>
```

4.3.4 Creating the group of Tacacs+ Servers

Create a group of Tacacs+ servers where server T1 is added. In this case, the group identifier is *GrupoTac.*

```
AAA config>group server tacacs+ GrupoTac
Tacacs+ group GrupoTac>server T1
Tacacs+ group GrupoTac>
```

4.3.5 Creating the authentication method list

Method lists that apply shell authentication are known as *authentication login*. They are created using the *AuthenLogin* identifier.

```
AAA config>authentication login AuthenLogin
Login list AuthenL...>
```

A *group* method is created for the group of recently created Radius servers.

```
Login list AuthenL...>method 1 group GrupoRad
Login list AuthenL...>
```

4.3.6 Creating the accounting method list

Method lists that apply accounting commands are known as *accounting commands*. They are created using the *AccCmds* identifier.

```
AAA config>accounting commands AccCmds
Cmds list AccCmds>
```

This accesses privilege level 15.

```
Cmds list AccCmds>privilege-level 15
Cmds lvl 15>
```

A *group* method is created for the group of recently created Tacacs+ servers.

```
Cmds lvl 15>method 1 group GrupoTac
Cmds lvl 15>
```

4.3.7 Enabling AAA

Before associating the method list to any service, enable the AAA feature.

```
AAA config>enable
```

```
AAA config>
```

4.3.8 Associating the method lists to the console

Access the console configuration menu.

```
Config>set console
-- Console configuration --
Con config>
```

Configure the authentication with the previously created method list, *AuthenLogin*.

```
Con config>login authentication AuthenLogin
Con config>
```

Configure *accounting* for level 15 commands with the previously created method list, *AccCmds*.

```
Con config>accounting commands 15 AccCmds
Con config>
```

The configuration is now complete. Now, when a user accesses the router shell through the console, he undergoes an authentication process. This process is validated by a Radius server. When the user is granted access, all level 15 commands executed by the user are registered.

The next console shows the full configuration:

```
  log-command-errors
   no configuration
   set data-link x25 serial0/1
   feature aaa
; -- AAA user configuration --
      enable
      tacacs-servers
         server "T1"
            port 4499
            key ciphered 0x660F5E0D4DD8714C
            host 192.168.1.12
         exit
;
      exit
;
      radius-servers
         server "R1"
            port 1881
            key ciphered 0xE13697A11E572446
            host 192.168.1.11
         exit
;
      exit
;
      group server tacacs+ "GrupoTac"
         server T1
      exit
;
      group server radius "GrupoRad"
         server R1
      exit
;
      authentication login "AuthenLogin"
         method 1 group GrupoRad
      exit
;
      accounting commands "AccCmds"
         privilege-level 15
            action-type start-stop
            method 1 group GrupoTac
         exit
;
      exit
```

```
;
   exit
;
   network ethernet0/0
; -- Ethernet Interface User Configuration --
      ip address 192.168.1.1 255.255.255.0
;
   exit
;
   set console
; -- Console configuration --
      login authentication AuthenLogin
      accounting commands 15 AccCmds
   exit
;
   dump-command-errors
   end
```