HTTP Protocol

bintec Dm737-I

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

Chapter 1 Introduction

1.1 HTTP Protocol

The Hyper Text Transfer Protocol (HTTP) is a web protocol (WWW) used in every transaction. The hypertext is the web page content, while the transfer protocol is the system used to send requests to access a website and receive responses from the same, thus producing the information you subsequently see on the screen.

Practical information systems require more functionality than the mere recovery of data, including search, update and annotation functions. HTTP allows an open-ended set of methods that indicate the purpose of a request. It is based on the use of a reference provided by the Uniform Resource Identifier (URI), as a location (URL) or name (URN), to indicate resources to which methods must be applied. Messages are passed in a similar format to that used by Internet Mail and Multipurpose Internet Mail Extensions (MIME).

HTTP is also used as a generic protocol for communication between clients or *proxies/gateways* and other Internet protocols such as SMTP, TNP, FTP, Gopher and WAIS. It enables basic *hypermedia* access to resources from diverse applications and simplifies client implementation.

Hyper Text Transfer Protocol Secure (HTTPS) is a combination of HTTP and SSL/TLS. It is used to ensure secure web connections, primarily by sites handling payment transactions or sensitive information. Communication is ciphered in the HTTP variant.

Web servers that use HTTP and /or HTTPS have been incorporated in some of our devices. These servers allow the protocols to be configured in graphic mode and do not require the use of an external program, other than a Web client (browser).

Chapter 2 Configuration

2.1 Configuration Commands

The following section describes the commands needed to configure HTTP.

Enter the following commands to access the HTTP configuration environment:

```
*config

Config>feature http

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

The following table summarizes the HTTP configuration commands:

Command	Function
? (HELP)	Lists the commands or their options.
AUTHORIZATION	Sets AAA authorization method list.
DISABLE	Disables HTTP.
LIST	Lists the HTTP configuration.
HOST-IGNORE	Ignores the hostname in HTTP requests.
HTTP-REDIRECT	Configures the number of HTTP file requests and which ones need to be redirected to other specific addresses.
LOGIN	Sets AAA authentication login methods list.
MAX-SESSIONS	Configures the maximum number of HTTP sessions.
MSS	Configures the maximum number of segments to use per TCP session.
NO	Configures the default value for the selected parameter.
RX-BUFF	Configures the reception buffer size.
SECURE-SERVER	Configures the HTTPS server.
TRACE-LEVEL	Configures the HTTP trace level to display (if enabled).
TX-BUFF	Configures the reception transmission size.
PORT	Configures the port assigned to the HTTP server.
EXIT	Returns to the previous prompt.

2.1.1 ?(HELP)

Entering **?** displays all the available commands. You can also use the **?** symbol to view the various options for each command.

Syntax:

```
HTTP config>?
```

Example:

```
HTTP config>?

  authorization    Set AAA authorization options
  disable          Disables HTTP server
  host-ignore      Ignore hostname on requests
  http-redirect    Configure parameters to redirect files to specified
                   locations
  list             List configuration
  login            Set AAA login options
  max-sessions     Max number of http sessions
  mss              Configures the maximum TCP segment size
  no               Set default configuration
  port             Set port number
  rx-buff          Configures the size of the buffers used to receive HTTP
  secure-server    Secure server (HTTPS) configuration
  trace-level      Configures the level of the traces to show
  tx-buff          Configures the size of the buffers used to transmit HTTP
  exit
```

2.1.2 DISABLE

Disables the device's HTTP server, preventing access to the server through the protocol.

Syntax:

```
HTTP config>disable
```

Example:

```
HTTP config>disable
HTTP config>
```

2.1.3 HOST-IGNORE

Disables the *hostname* check in an HTTP request.

Syntax:

```
HTTP config>host-ignore
```

Example:

```
HTTP config>host-ignore
HTTP config>
```

2.1.4 HTTP-REDIRECT

This command tells the HTTP server when to apply redirect to HTTP queries.

```
HTTP config>http-redirect ?
  access-list    Configure the access-list for the redirection of files
  file           Assign a location to a specified file from where it must be
                 downloaded
HTTP config>
```

2.1.4.1 HTTP-REDIRECT ACCESS-LIST

Assigns an access control list to specify certain ranges of source IP addresses to which redirect is applied if necessary.

```
HTTP+ redirect access-list ?
  <1..99>    Value in the specified range
HTTP config>
```

2.1.4.2 HTTP-REDIRECT FILE

Assigns one or more firmware files to the new location the browser must redirect to (to download the files in question). This new location is broadcast by the HTTP server in response to a query.

Syntax:

```
HTTP config>http-redirect file <files> <new URL>
```

Example:

```
HTTP config>http-redirect file jquery http://wwww.repository.es/private/
```

2.1.5 LIST

Run **list** to view the content of the HTTP configuration.

Syntax:

```
HTTP config>list
```

Example:

List corresponding to the default configuration:

```
HTTP config>list

HTTP server: ENABLED
HTTP port:       80

HTTP MSS: 1460
HTTP Rx Buffer:  2048
HTTP Tx Buffer:  8192

HTTP Max sessions:       10
HTTP trace level:        warning
HTTP config>
```

2.1.6 LOGIN

Associates an *authentication login* methods list defined using the AAA feature. This way, the HTTP service applies the methods from the associated list when it needs to authenticate.

Syntax:

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

Example:

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

In this example, the *AuthLogin* methods list has been configured so it can be used to authenticate user access to the WEB.

Method lists can only be applied if the AAA feature is enabled. To do this, once you have configured the AAA, enable it to apply the lists to the various services. For further information on how to configure the AAA feature, please see manual “Dm800-I AAA Feature”.

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.

2.1.7 AUTHORIZATION

Associates an *authorization* method list defined using the AAA feature. This way, the HTTP service applies the methods from the associated list when it needs to authorize.

Syntax:

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

Example:

```
HTTP config>authorization exec AuthUser
HTTP config>
```

In this example, the *AuthUser* methods list has been configured so it can be used to authorize a user with the corresponding access-level to the WEB.

Method lists can only be applied if the AAA feature is enabled. To do this, once you have configured the AAA, enable it to apply the lists to the various services. For further information on how to configure the AAA feature, please see manual “Dm800-I AAA Feature”.

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.
11.01.00	This command was introduced as of version 11.01.00.

2.1.8 MAX-SESSIONS

Configures the maximum number of HTTP sessions that can be simultaneously active. Bear in mind that a browser usually opens multiple sessions (between four and five) at the same time. Once the maximum number of simultaneously active sessions has been reached, further connection attempts are rejected until one of the established sessions has been released.

Syntax:

```
HTTP config>max-sessions <number-of-sessions>
```

Example:

Configuring the number of sessions to 20:

```
HTTP config>max-sessions 20
HTTP config>
```

2.1.9 MSS

Configures the Maximum Segment Size (*MSS*) to use per TCP session. This value is between 512 and 4096.

Syntax:

```
HTTP config>mss <value>
```

Example:

Configuring the segment size to 1024 bytes:

```
HTTP config>mss 1024
HTTP config>
```

2.1.10 NO

Run **no** to undo a command action or to restore a parameter's default value.

2.1.10.1 NO PORT

Sets the default value of the port parameter assigned to the device's HTTP server.

Syntax:

```
HTTP config>no port
```

Example:

```
HTTP config>no port
HTTP config>
```

2.1.10.2 NO DISABLE

Enables HTTP.

Syntax:

```
HTTP config>no disable
```

Example:

```
HTTP config>no disable
HTTP config>
```

2.1.10.3 NO HTTP-REDIRECT ACCESS LIST

Deletes the assignment of an access control list (ACL) from the HTTP protocol. This reallocation implies that the server will redirect file queries where necessary.

Syntax:

```
HTTP config>no http-redirect access-list
```

2.1.10.4 NO HTTP-REDIRECT FILE

Prevents specific files from being relocated to a new location.

Syntax:

```
HTTP config>no http-redirect file <files>
```

2.1.10.5 NO LOGIN AUTHENTICATION

Eliminates the associated *authentication login* methods list.

Syntax:

```
HTTP config>no login authentication
```

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.

2.1.10.6 NO AUTHORIZATION EXEC

Eliminates the associated *authorization* method list.

Syntax:

```
HTTP config>no authorization exec
```

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.
11.01.00	This command was introduced as of version 11.01.00.

2.1.10.7 NO SECURE-SERVER CA

Eliminates the certification authority certificate supported by your computer.

Syntax:

```
HTTP config>no secure-server ca <ca_name>
```

Example:

```
HTTP config>no secure-server ca ROUTER_A.CER
HTTP config>
```

2.1.10.8 NO SECURE-SERVER DONT-VERIFY

Allows the HTTPS server to request a certificate from clients connecting to it.

Syntax:

```
HTTP config>no secure-server dont-verify
```

Example:

```
HTTP config>no secure-server dont-verify
HTTP config>
```

2.1.10.9 NO SECURE-SERVER ENABLE

Disables the HTTPS server.

Syntax:

```
HTTP config>no secure-server enable
```

2.1.10.10 NO SECURE-SERVER PORT

Eliminates the port established for the HTTPS server.

Syntax:

```
HTTP config>no secure-server port <port>
```

Example:

```
HTTP config>no secure-server port 443
HTTP config>
```

2.1.10.11 NO SECURE-SERVER SSLV2

Disables SSLv2 compatibility.

Syntax:

```
HTTP config>no secure-server sslv2
```

Example:

```
HTTP config>no secure-server sslv2
HTTP config>
```

Command history:

Release	Modification
11.01.10	This command was obsoleted as of version 11.01.10.

2.1.10.12 NO SECURE-SERVER SSLV3

Disables SSLv3 compatibility.

Syntax:

```
HTTP config>no secure-server sslv3
```

Example:

```
HTTP config>no secure-server sslv3
HTTP config>
```

Command history:

Release	Modification
11.01.10	This command was introduced as of version 11.01.10.

2.1.10.13 NO SECURE-SERVER USER

Eliminates the HTTPS server user certificate.

Syntax:

```
HTTP config>no secure-server user <cert>
```

Example:

```
HTTP config>no secure-server user ROUTER.CER
HTTP config>
```

2.1.11 PORT

Configures the port assigned to the device's HTTP server.

Syntax:

```
HTTP config>port <port_id>
```

Example:

```
HTTP config>port 80
HTTP config>
```

2.1.12 RX-BUFF

Configures the reception window size used for the TCP session. This value can range from 2048 to 65534.

Syntax:

```
HTTP config>rx-buff <size>
```

Example:

```
HTTP config>rx-buff 2048
HTTP config>
```

2.1.13 SECURE-SERVER

Configures the HTTPS parameters available in the device.

2.1.13.1 SECURE-SERVER CA <ca-name>

Configures the secure certificate authorities for the TLS sessions established for the HTTPS protocol. Multiple certificates from different certificate authorities can be configured. The certificates must be preloaded onto the computer.

The certification authorities allow the certificates to be validated on the devices that communicate with the router through TLS.

To load a certificate in base64 in the device from the IPSec certificates menu, execute **certificate <certname> base64** and enter the certificate. This generates a configuration in IPSec as shown in the example. For further details on this, please see the section on *Certificates* in *Chapter* 2 of the *bintec Dm739-I IPSEC* manual.

Syntax:

```
HTTP config>secure-server ca <ca-name>
```

Example:

Loading a root certificate in the bintec example using the **certificate <name> base64** command from the **protocol ip>ipsec>cert** menu.

```
CERTIFICATES config$certificate SAMPLECA.CER base64

Introduce the Certificate (Base 64 format)
Enter <cr> to escape
-----BEGIN CERTIFICATE-----
MIIEmzCCA4OgAwIBAgIJAIjCeKBqciDFMA0GCSqGSIb3DQEBBAUAMIGPMQswCQYD
VQQGEwJTUDEPMA0GA1UECBMGTWFkcmlkMRQwEgYDVQQHEwtUcmVzIENhbnRvczEU
MBIGA1UEChMLVGVsZGF0IFMuQS4xGzAZBgNVBAsTEklQIFRlbGVwaG9ueSBHcm91
cDEmMCQGA1UEAxMdVGVkYXQgQ2VydGlmaWNhdGlvbiBBdXRob3JpdHkwHhcNMDcw
NjExMTUxNDE2WhcNMTcwNjA4MTUxNDE2WjCBjzELMAkGA1UEBhMCU1AxDzANBgNV
BAgTBk1hZHJpZDEUMBIGA1UEBxMLVHJlcyBDYW50b3MxFDASBgNVBAoTC1RlbGRh
dCBTLkEuMRswGQYDVQQLExJJUCBUZWxlcGhvbnkgR3JvdXAxJjAkBgNVBAMTHVRl
ZGF0IENlcnRpZmljYXRpb24gQXV0aG9yaXR5MIIBIjANBgkqhkiG9w0BAQEFAAOC
AQ8AMIIBCgKCAQEAmSRtZ9wHCksPAzkdMvqYyUAnOecJWw/Aai67TObXhi/a4w5T
Onbf8LKjsGWamksMU6p7iv7n4rd6Kqyr1q/S1yP9XfENiVfsmu3dq9ehkipg5ixw
E16xAdpGXJpdob8zOkUwiKaJib8LsTE38upaA2iV++bQSIMKcma4rnlPW1wn9jAJ
```

```
mMwTMKCT7vT7OfcEIVzB7P1RW9phTMmQsSTTg7SMlRxTN0c2WW216aLOO5qRwvt4
xzcoXRVYbm2aBj7LucjsOrgoEdscmga8kK7PYdetxqti1n6RfjP2BXmAUrKh91c3
61fazv+pNxpKSL0hQ8Gb+hUxPyjZJTTW+Zih+wIDAQABo4H3MIH0MB0GA1UdDgQW
BBQsJNVrUzOnr7Rxj4FfdiBLKOSv9DCBxAYDVR0jBIG8MIG5gBQsJNVrUzOnr7Rx
j4FfdiBLKOSv9KGBlaSBkjCBjzELMAkGA1UEBhMCU1AxDzANBgNVBAgTBk1hZHJp
ZDEUMBIGA1UEBxMLVHJlcyBDYW50b3MxFDASBgNVBAoTC1RlbGRhdCBTLkEuMRsw
GQYDVQQLExJJUCBUZWxlcGhvbnkgR3JvdXAxJjAkBgNVBAMTHVRlZGF0IENlcnRp
ZmljYXRpb24gQXV0aG9yaXR5ggkAiMJ4oGpyIMUwDAYDVR0TBAUwAwEB/zANBgkq
hkiG9w0BAQQFAAOCAQEAR16Gjs16Sqz04v/RJeRb+fcbKvAgzO3sWpUyYwzU/j6L
7R5XVbgimX4FQ3qxnrNeYXCTtZAM8yMWKpnX1d9ZDgGqZsOV0NrjlSGAYk3yvdM5
cNEXQpLDkKhjN8ageD48yNWpBTzbTDk/jQXCfktF3L93qpB/W76taC54bb1LojHs
kcPXB4pzgN7QGct/wVyg2KNcMaQITmOesY+Qqt8T0QxZomsn8ldz6c7HAoRurmnB
x/SCdpqfwMMnS7ap/5y+uPNuROw3ib8GWWqq6I3/bUqxgkgEwWD8OdkYHKNJV5h8
0zJjXH5/jqf1hmwKV07QQ+WxENxdtc6FB3Idmgj33w==
-----END CERTIFICATE-----
```

The certificate data can be viewed through the command found in the same menu. If you have loaded from the static configuration, save and reboot before loading the certificate in the memory.

```
CERTIFICATES config$list loaded-certificates
----------------------- SAMPLECA.CER (from config)
 Subject:
  CN (Common Name        ): Sample Certification Authority
  OU (Organizational Unit): IP Telephony Group
  O  (Organization Name  ): Sample S.A.
  L  (Locality           ): Tres Cantos
  S  (State or Province  ): Madrid
  C  (Country Name       ): SP

 Issuer: A:SAMPLECA.CER

CERTIFICATES config$
```

This will generate the following configuration in the certificates menu. This configuration can be used directly in any other device without having to reload the certificate in base64.

```
protocol ip
; -- Internet protocol user configuration –
   Ipsec
; -- IPSec user configuration --
      Cert
; -- Cert user configuration --
         file new SAMPLECA.CER
         file add 0x3082049B30820383A00302010202090088C278A06A7220C5300D06092A864886
         file add 0xF70D01010405003081 8F310B30090603550406130253503 10F300D0603550408
         file add 0x13064D616472696431143012060355040713 0B547265732043616E746F733114
         file add 0x3012060355040A130B54656C64617420532E412E311B3019060355040B131249
         file add 0x502054656C6570686F6E792047726F7570312630240603550403131D54656461
         file add 0x742043657274696669636174696F6E20417574686F72697479301E170D303730
         file add 0x36313131353134313 65A170D31373036303831353134313 65A30818F310B3009
         file add 0x0603550406130253503 10F300D060355040813064D61647269643111430120603
         file add 0x550407130B547265732043616E746F7331143012060355040A130B54656C6461
         file add 0x7420532E412E311B3019060355040B131249502054656C6570686F6E79204772
         file add 0x6F7570312630240603550403131D546564617420436572746966696361746 96F
         file add 0x6E20417574686F7269747930820122300D06092A864886F70D01010105000382
         file add 0x010F003082010A0282010100992 46D67DC070A4B0F03391D32FA98C9402739E7
         file add 0x095B0FC06A2EBB4CE6D7862FDAE30E533A76DFF0B2A3B0659A9A4B0C53AA7B8A
         file add 0xFEE7E2B77A2AACABD6AFD2D723FD5DF10D8957EC9AEDDDABD7A1922A60E62C70
         file add 0x135EB101DA465C9A5DA1BF333A453088A68989BF0BB13137F2EA5A036895FBE6
         file add 0xD048830A7266B8AE794F5B5C27F6300998CC1330A093EEF4FB39F704215CC1EC
         file add 0xFD515BDA614CC990B124D383B48C951C53374736596DB5E9A2CE3B9A91C2FB78
         file add 0xC737285D15586E6D9A063ECBB9C8EC3AB82811DB1C9A06BC90AECF61D7ADC6AB
         file add 0x62D67E917E33F605798052B2A1F75737EB57DACEFFA9371A4A48BD2143C19BFA
         file add 0x15313F28D92534D6F998A1FB0203010001A381F73081F4301D0603551D0E0416
         file add 0x04142C24D56B5333A7AFB4718F815F76204B28E4AFF43081C40603551D230481
         file add 0xBC3081B980142C24D56B5333A7AFB4718F815F76204B28E4AFF4A18195A48192
         file add 0x30818F310B30090603550406130253503 10F300D060355040813064D61647269
         file add 0x64311430120603550407130B547265732043616E746F7331143012060355040A
         file add 0x130B54656C64617420532E412E311B3019060355040B131249502054656C6570
         file add 0x686F6E792047726F7570312630240603550403131D5465646174204365727469
         file add 0x6669636174696F6E20417574686F7269747982090088C278A06A7220C5300C06
         file add 0x03551D13040530030101FF300D06092A864886F70D0101040500038201010047
         file add 0x5E868ECD7A4AACF4E2FFD125E45BF9F71B2AF020CCEDEC5A9532630CD4FE3E8B
         file add 0xED1E5755B822997E05437AB19EB35E617093B5900CF323162A99D7D5DF590E01
         file add 0xAA66C395D0DAE3952180624DF2BDD33970D1174292C390A86337C6A0783E3CC8
         file add 0xD5A9053CDB4C393F8D05C27E4B45DCBF77AA907F5BBEAD682E786DBD4BA231EC
         file add 0x91C3D7078A7380DED019CB7FC15CA0D8A35C31A4084E639EB18F90AADF13D10C
         file add 0x59A26B27F25773E9CEC702846EAE69C1C7F482769A9FC0C3274BB6A9FF9CBEB8
         file add 0xF36E44EC3789BF06596AAAE88DFF6D4AB1824804C160FC39D9181CA34957987C
         file end 0xD332635C7E7F8EA7F5866C0A574ED043E5B110DC5DB5CE8507721D9A08F7DF
;
         certificate SAMPLECA.CER load
      exit
;
   exit
;
exit
```

Finally, configure the certificate as a secure certificate authority for HTTPS TLS connections.

```
feature http
   secure-server ca SAMPLECA.CER
exit
```

2.1.13.2 SECURE-SERVER CIPHERS

Configures the cipher and authentication algorithms negotiated in TLS connections.

The string format is the same as the one used by openssl. To see the format and examples, please visit the following website: *http://www.openssl.org/docs/apps/ciphers.html* .

Syntax:

```
HTTP config>secure-server ciphers <string>
```

2.1.13.3 SECURE-SERVER DONT-VERIFY

This command stops the device from requesting an X509 certificate from connected clients (clients are not authenticated). Default is disabled, which means the device does request certification from clients and checks that they are signed by a secure certificate authority.

Syntax:

```
HTTP config>secure-server dont-verify
```

2.1.13.4 SECURE-SERVER ENABLE

Enables the device's HTTPS server, allowing access to the server through the protocol.

Syntax:

```
HTTP config>secure-server enable
```

Example:

```
HTTP config>secure-server enable
HTTP config>
```

2.1.13.5 SECURE-SERVER PORT

Configures the port assigned to the device's HTTPS server.

Syntax:

```
HTTP config>secure-server port <port_id>
```

Example:

```
HTTP config>secure-server port 443
HTTP config>
```

2.1.13.6 SECURE-SERVER SSLV2

SSLv2 connections are supported when using this command.

Syntax:

```
HTTP config> secure-server sslv2
```

Command history:

Release	Modification
11.01.10	This command was obsoleted as of version 11.01.10.

2.1.13.7 SECURE-SERVER SSLV3

SSLv3 connections are supported when using this command.

Syntax:

```
HTTP config> secure-server sslv3
```

Command history:

Release	Modification
11.01.10	This command was introduced as of version 11.01.10.

2.1.13.8 SECURE-SERVER USER

Configures the certificate used by the device when it is behaving as the TLS connections server. The certificate must be preloaded onto the device and have a corresponding private key.

For more information on how to generate private keys and load signed certificates onto the router, please see manual *bintec Dm739-I-IPSEC*.

Syntax:

```
HTTP Config>secure-server user <cert-name>
```

Example:

Generating a private key and a Certificate Signing Request (CSR):

```
IPSec config>key rsa generate user.csr 512
RSA Key Generation.
Please, wait for a few seconds.
 RSA Key Generation done.
Checking..OK
Key Generation Process Finished.
Generate CSR?
(Yes/No)? y
Common Name        :  []? Sample User Certificate
Country            :  []? SP
Locality           :  []? Tres Cantos
State or Province  :  []? Madrid
Organization       :  []? Sample S.A.
Organizational Unit:  []? IP Telephony Group
E-mail             :  []?
RSA Signature(MD5/SHA1/MD2): [md5]?
Save in file(Yes/No)? y
File Name: []? sample.csr
File Name: [A:USER.CSR]? y
CSR saved.
Do not forget to save RSA keys.
IPSec config>
```

Lists the encrypted private key generated. Listing the user.csr file gives us the CSR the certification authority must sign:

```
IPSec config>sho conf
; Showing Menu and Submenus Configuration for access-level 15 ...
; XXX NOE Router 2 226 Version 10.7.7

         key rsa file add 0x341DC332F23BD75492E8583A82F10A8CFCA4349F531563EF
         key rsa file add 0xCA002248A79CFCFE24FBBCE0FA9C3BF99B02C316C9C5EB44
         key rsa file add 0xA2630B28F04183766A79C93BD967380425955159D32B7035
         key rsa file add 0x448A8DB2954FA8E53132CDAFE7365FED6BAE5CA55ED8809E
         key rsa file add 0x89191F4762B0850603BE8A61AFD3CC786EC5ED1EB08F191C
         key rsa file add 0xCF7B8FD6AF0C37BDF33BEC201C6B58B1FAC419DF8F68F525
         key rsa file add 0x31F285C22AD9896587FE9095A8355C6F35075CDFAE7E2485
         key rsa file add 0x6E75FC669194A00DED45B8AFDF3B99B6A162F7FE14B0F3B8
         key rsa file add 0xDC0E4D442F9EC916D05F161BABA2D3D803AABADF64A8E6EC
         key rsa file add 0x1CBD2973DC158D77A872B75FE4E99277E877E49C117FC6D9
         key rsa file add 0xC8E29F6D1D84030B955E1A6A15E2F15386CA5F6598148876
         key rsa file add 0x0C27B15CF5D812C2922706CF25C7D42DE09DCB4330125C2B
         key rsa file add 0x911BC4C084B9ADE1D6D5B7DDDDD030692F2EC9E66E0E7D74
         key rsa file add 0x782F99AC347A1BCAC42CEBCEF011F1D25646465BED83ABE9
         key rsa file add 0xBCC82DFBE5BA79E4D024AA7F6AB05D03101335AD37882784
         key rsa file add 0xF5F01429118037178894D1823871D8F498F9B2B5C1EB488D
         key rsa file add 0x9D6349C59E11A617F5622FFC33D8D6272D7C13C6F9B494CE
         key rsa file add 0x3148B09CB12A6B438EC87E272FBC4A0C629CD9DDCAC1B9A3
         key rsa file add 0x8DFEC5C76588A09F6417A94ACF76313C89198FE0D7B5E1B2
         key rsa file add 0x02249303A5A3731A0162B207950C41E18A3952DC84415084
         key rsa file add 0x41E09EF3908BD169243C9A611D1EA318FCEF7A2BD0378108
         key rsa file add 0xFD2E886FE114EAFBB1F18892F67FEA2173D8F05B5ED54B67
         key rsa file add 0x0D649530B2392230C9AA2D9974777147DFBCCD2067222A11
         key rsa file add 0x8C49A3D60E22901AC5103313CE5CC0B9FA1A2F1607BC55EE
         key rsa file add 0xA6EB05FB527E786CD4529F1388F6E66AFBFA41234902488E
         key rsa file end 0xB4303ABF65069D25D17145D8695CBD88EC0C92EECC210B36
IPSec config>
IPSec config>exit
IP config>exit
Config>file type a:user.csr
-----BEGIN CERTIFICATE REQUEST-----
MIIBRDCB7wIBADCBiTEgMB4GA1UEAxMXU2FtcGxlIFVzZXIgQ2VydGlmaWNhdGUx
CzAJBgNVBAYTAlNQMRQwEgYDVQQHEwtUcmVzIENhbnRvczEPMA0GA1UECBMGTWFk
cmlkMRQwEgYDVQQKEwtUZWxkYXQgUy5BLjEbMBkGA1UECxMSSVAgVGVsZXBob255
IEdyb3VwMFwwDQYJKoZIhvcNAQEBBQADSwAwSAJBANflczMh4//+vDYBHSrzou5N
LamVxi4GStHWJatFHdsKfiYU7S6DKjgmB9Acyi2haH+bydDqD2pMDCPMvfyURjEC
AwEAAaAAMA0GCSqGSIb3DQEBBAUAA0EAEzLTh/ZwLKM2J9IyEeYqCevSFm/zye53
bz1R58go44cpLWgT9YL3kZoKrKAqevWdNRyjHKuwZmt+XoHaRVkSog==
-----END CERTIFICATE REQUEST-----
```

Once the certificate authority has signed the CSR, the user certificate can be loaded into the router in a similar way to the previous example. Finally, configure the certificate to use HTTPS as the user certificate. The final configuration (which can also be used directly in another device without repeating the whole process) is as follows:

```
protocol ip
; -- Internet protocol user configuration --
   ipsec
; -- IPSec user configuration --
      key rsa file add 0x341DC332F23BD75492E8583A82F10A8CFCA4349F531563EF
      key rsa file add 0xCA002248A79CFCFE24FBBCE0FA9C3BF99B02C316C9C5EB44
      key rsa file add 0xA2630B28F04183766A79C93BD967380425955159D32B7035
      key rsa file add 0x448A8DB2954FA8E53132CDAFE7365FED6BAE5CA55ED8809E
      key rsa file add 0x89191F4762B0850603BE8A61AFD3CC786EC5ED1EB08F191C
      key rsa file add 0xCF7B8FD6AF0C37BDF33BEC201C6B58B1FAC419DF8F68F525
      key rsa file add 0x31F285C22AD9896587FE9095A8355C6F35075CDFAE7E2485
      key rsa file add 0x6E75FC669194A00DED45B8AFDF3B99B6A162F7FE14B0F3B8
      key rsa file add 0xDC0E4D442F9EC916D05F161BABA2D3D803AABADF64A8E6EC
      key rsa file add 0x1CBD2973DC158D77A872B75FE4E99277E877E49C117FC6D9
      key rsa file add 0xC8E29F6D1D84030B955E1A6A15E2F15386CA5F6598148876
      key rsa file add 0x0C27B15CF5D812C2922706CF25C7D42DE09DCB4330125C2B
      key rsa file add 0x911BC4C084B9ADE1D6D5B7DDDDD030692F2EC9E66E0E7D74
      key rsa file add 0x782F99AC347A1BCAC42CEBCEF011F1D25646465BED83ABE9
      key rsa file add 0xBCC82DFBE5BA79E4D024AA7F6AB05D03101335AD37882784
      key rsa file add 0xF5F01429118037178894D1823871D8F498F9B2B5C1EB488D
      key rsa file add 0x9D6349C59E11A617F5622FFC33D8D6272D7C13C6F9B494CE
      key rsa file add 0x3148B09CB12A6B438EC87E272FBC4A0C629CD9DDCAC1B9A3
      key rsa file add 0x8DFEC5C76588A09F6417A94ACF76313C89198FE0D7B5E1B2
      key rsa file add 0x02249303A5A3731A0162B207950C41E18A3952DC84415084
      key rsa file add 0x41E09EF3908BD169243C9A611D1EA318FCEF7A2BD0378108
      key rsa file add 0xFD2E886FE114EAFBB1F18892F67FEA2173D8F05B5ED54B67
      key rsa file add 0x0D649530B2392230C9AA2D9974777147DFBCCD2067222A11
      key rsa file add 0x8C49A3D60E22901AC5103313CE5CC0B9FA1A2F1607BC55EE
      key rsa file add 0xA6EB05FB527E786CD4529F1388F6E66AFBFA41234902488E
      key rsa file end 0xB4303ABF65069D25D17145D8695CBD88EC0C92EECC210B36
      cert
; -- Cert user configuration --
         file new USER.CER
         file add 0x308203DB308202C3A003020102020101300D06092A864886F70D010104050030
         file add 0x818F310B3009060355040613025350310F300D060355040813064D6164726964
         file add 0x311430120603550407130B547265732043616E746F7331143012060355040A13
         file add 0x0B54656C64617420532E412E311B3019060355040B131249502054656C657068
         file add 0x6F6E792047726F7570312630240603550403131D546564617420436572746966
         file add 0x69636174696F6E20417574686F72697479301E170D30373036313131353323731
         file add 0x335A170D3137303630383135323731335A3073310B3009060355040613025350
         file add 0x310F300D060355040813064D61647269643114301206035504A130B54656C64
         file add 0x617420532E412E311B3019060355040B131249502054656C6570686F6E792047
         file add 0x726F75703120301E0603550403131753616D706C652055736572204365727469
         file add 0x666963617465305C300D06092A864886F70D0101010500034B003048024100D7
         file add 0xE5733321E3FFFEBC36011D2AF3A2EE4D2DA995C62E064AD1D625AB451DDB0A7E
         file add 0x2614ED2E832A382607D01CCA2DA1687F9BC9D0EA0F6A4C0C23CCBDFC94463102
         file add 0x03010001A38201233082011F30090603551D1304023000302C06096086480186
         file add 0xF842010D041F161D4F70656E53534C2047656E657261746564204365727466966
         file add 0x6963617465301D0603551D0E0416041423F7EC38E8281BE6E95D1C4D09DFB04D
         file add 0x53909A1F3081C40603551D230481BC3081B980142C24D56B5333A7AFB4718F81
         file add 0x5F76204B28E4AFF4A18195A4819230818F310B3009060355040613025350310F
         file add 0x300D060355040813064D616472696431143012060355040713 0B547265732043
         file add 0x616E746F7331143012060355040A130B54656C64617420532E412E311B301906
         file add 0x0355040B131249502054656C6570686F6E792047726F75703126302406035504
         file add 0x03131D546564617420436572746966696361746696F6E20417574686F72697479
         file add 0x82090088C278A06A7220C5300D06092A864886F70D01010405000382010100 7A
         file add 0xABF460D76CEECA2C4B6AE2203F9A1546B6DC646DC54F3D92D8EE57371496AA89
         file add 0x170A6BF836D7A40BD608480B73D53F8C38B27C110532B1B2B8E0967F3BB67DA3
         file add 0x7503EB26B08417D17246190E1D0584F325C1CA691748730AF1B1B9EB6E8F06B6
         file add 0x85F8A4600927F5F68E08D042DEBE97E62500C3D4AE19A3302B085FF572D0E5C8
         file add 0x68C56B6D1923EE9E33A50222A9D48A0515B44C2D324C6CDFB8E1B0F3D5E56FC1
         file add 0xEF618B5898E613E4EFC194D782B8241C1267523A6D8A02449D6AA07A609D4279
         file add 0x1739E27DA61804160075C9617E8D5C585A8F0E0400DD2650FC372EE8F7B22F3D
         file end 0xEEA5B7701DD27E44870E49F1486930B28E6D45874AA62D3F4F1BEFA4EAEF84
;
         certificate USER.CER load
      exit
;
   exit
;
exit
;
feature http
   secure-server user USER.CER
exit
```

2.1.14 TRACE-LEVEL

Configures the trace level you want to show when the HTTP events linked to the Web server are enabled. The minimum level is *error* and the maximum level is *msg-dump.*

- error: messages produced by serious errors that can cause the HTTP server to stop operating.
- warning: messages caused by unexpected errors when the HTTP server is operating.
- info: traces relating to normal HTTP server operations.
- debug: debugging traces. Traces the HTTP server operations in greater detail.
- msg-dump: displays traces related to the messages exchanged when the HTTP server is operating.

Each level includes the ones above. If, for example, you enable the debug traces, the info, warning and error traces levels will also be activated.

Syntax:

```
HTTP config>trace-level ?
  msg-dump     trace level for http
  debug        trace level for http
  info         trace level for http
  warning      trace level for http
  error        trace level for http
```

Example:

```
HTTP config>trace-level warning
HTTP config>
```

2.1.15 TX-BUFF

Configures the size of the transmission window used by the TCP session. This value can range from 2048 to 65534.

Syntax:

```
HTTP config>tx-buff <size>
```

Example:

```
HTTP config>tx-buff 2048
HTTP config>
```

2.1.16 EXIT

Run **exit** to quit the HTTP configuration menu.

Syntax:

```
HTTP config>exit
```

Example:

```
HTTP config>exit
Config>
```

Chapter 3 Monitoring

3.1 Monitoring Commands

This section describes the commands used to monitor the HTTP protocol.

Enter the following commands to access the HTTP monitoring environment:

```
*monitor
Console Operator

+feature http
-- HTTP server user console --

HTTP+
```

The following table summarizes the HTTP monitoring commands:

Command	**Function**
? (HELP)	Lists the available commands or their options.
CACHE	Options relative to the HTTP server files cache.
EXIT	Returns to the previous prompt.

3.1.1 ? (HELP)

By entering ?, all the available commands are displayed. You can also use the ? symbol to view the various options for each command.

Syntax:

```
HTTP+?
```

Example:

```
HTTP+?
  cache    server cache options
  exit
HTTP+
```

3.1.2 CACHE

Options relevant to the HTTP server files cache.

3.1.2.1 CACHE CLEAR

Deletes all the HTTP server files cache. The first time a file is requested, the HTTP server reads the disk and saves it to RAM. Subsequent file requests are taken directly from the searched copy. This command is useful when you load new firmware that modifies the HTTP server pages.

Syntax:

```
HTTP+cache clear
```

3.1.3 EXIT

Exits the HTTP monitoring menu.

Syntax:

```
HTTP+exit
```

Chapter 4 Annex A

4.1 Third Party Software

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

Please see a copy of the OpenSSL license below:

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

OpenSSL License

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

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

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

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

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

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

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

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

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

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

Original SSLeay License:

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

All rights reserved.

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

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

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

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

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