Release Notes
System Software
Release 6.3.1
X-Generation

March 2003

System Software Release 6.3.1

This document describes the new features, changes, bugfixes and known issues in System Software Release 6.3.1.

1 Important Information

It is not possible to directly downgrade from System Software Release 6.3.1 to System Software Release 6.2.2. A staged downgrade, however, is possible:

- ➤ From System Software Release 6.3.1 downgrade to System Software 6.2.5.
- ➤ Save your configuration by entering `cmd=save` in the SNMP shell.
- ➤ Downgrade further to System Software Release 6.2.2 or earlier.

2 New Features

System Software Release 6.3.1 is a major new release of our system software, and it contains a number of important new features:

- SCEP (Simple Certificate Enrollment Protocol) (chapter 2.1, page 8)
- RADIUS Preset IPSec Peers (chapter 2.2, page 12)
- DSL Trace (chapter 2.3, page 16)
- QoS in a LAN (chapter 2.4, page 17)
- Software MPPC (chapter 2.5, page 17)
- MPPE with Hifn (chapter 2.6, page 18)
- TCP MSS Clamping for Non-NAT Interfaces (chapter 2.7, page 18)
- Bridging for Frame Relay (chapter 2.8, page 19)
- X.25 to TCP Gateway (chapter 2.9, page 19)
- `show rev` (chapter 2.10, page 27)

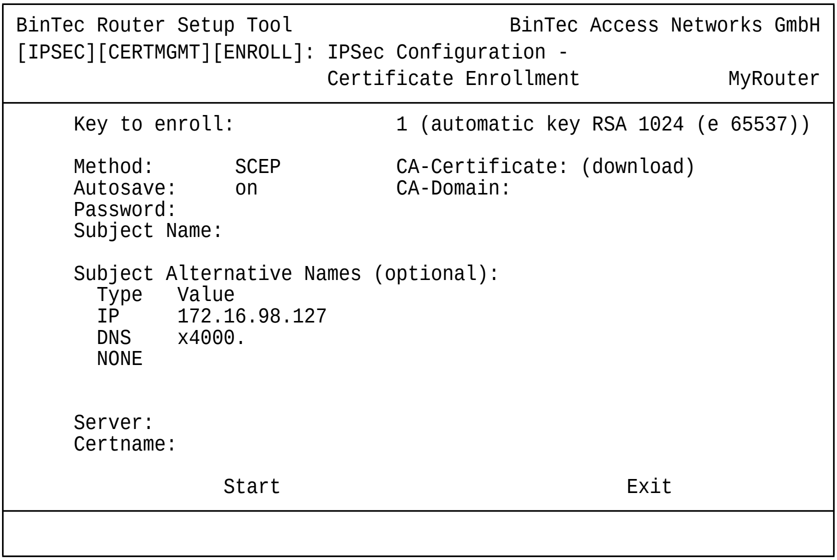
2.1 SCEP (Simple Certificate Enrollment Protocol)

Peer authentication through certificates in a Public Key Infrastructure is an important part of IPSec. Up to now certificates for a public key had to be manually enrolled and then downloaded in an additional step. BinTec routers so far have offered a mechanism to automatically create a PKCS #10 certificate request, but the certificate download was not automated. By implementing SCEP (Simple Certificate Enrollment Protocol), BinTec further simplifies the enrollment and download of public key certificates.

SCEP combines certificate enrollment and certificate download, and if the CA (Certificate Authority) servers support completely automated certificate distribution, the certificate can be retrieved immediately. If your CA does not support this degree of automation, the enrollment remains pending until the certificate has been signed by the CA and is available for download.

SCEP management is based on a newly designed MIB table, **CertMgmtTable**, but the use of SCEP can be configured through the Setup Tool, too. The menu ***IPSec*** ➡ ***Certificate and Key Management*** ➡ ***Key Management*** ➡ ***Request Cert*** has been changed accordingly.

It now looks like this (the example shows a 1024 bit RSA key generated by the IPSec Wizard that is to be enrolled):



```
BinTec Router Setup Tool                   BinTec Access Networks GmbH
[IPSEC][CERTMGMT][ENROLL]: IPSec Configuration -
                           Certificate Enrollment             MyRouter

     Key to enroll:               1 (automatic key RSA 1024 (e 65537))

     Method:       SCEP           CA-Certificate: (download)
     Autosave:     on             CA-Domain:
     Password:
     Subject Name:

     Subject Alternative Names (optional):
       Type   Value
       IP     172.16.98.127
       DNS    x4000.
       NONE


     Server:
     Certname:

                  Start                              Exit
```

The menu has the following additional fields (compared to the menu layout for manual certificate enrollment of older system software releases):

Field	Meaning
Method	Here you can choose between *Manual* enrollment and *SCEP*. The default value is *SCEP*. If you choose manual enrollment, the menu changes and looks as described in the IPSec manual. Only if you choose *SCEP* will the fields described below appear.
CA Certificate	Choose the CA certificate of the CA you are going to request the certificate from. If no CA certificates are available, the router will first download the CA certificate of the CA in question. It then continues the enrollment process unless essential parameters are still missing. In this case, it returns to the ***REQUEST CERT*** menu. In case the CA certificate does not contain a CRL or a CRL distribution point and no certificate server is configured on the router, the variable **NoCRLs** is set to *true*. Certificates from this CA will not be checked for validity.
Autosave	If you activate this option, the router will automatically save the enrollment process in its various states. This is useful if the enrollment can not be immediately completed or the router needs to be rebooted. If the state was not saved, the enrollment could not be completed. As soon as the enrollment is completed and the certificate has been downloaded from the CA server, it is automatically saved into the configuration of the router. Available choices are *on* and *off*.

Field	Meaning
CA Domain	Enter the domain name of the CA server the enrollment is sent to, e.g. ***enroll.ca.com***.
Password	In order to obtain certificates for your keys, you may need a password from the CA. Enter the password you have received from your CA here.
Certname	This replaces the field **Filename** from the menu for manual certificate enrollment. Enter a name for the resulting certificate.

Table 2-1: ***IPSEC*** ➜ ***CERT. AND KEY MNGMT.*** ➜ ***KEY MNGMT.*** ➜ ***REQUEST CERT*** (SCEP)

The menu ***IPSEC*** ➜ ***CERTIFICATE AND KEY MANAGEMENT*** ➜ ***KEY MANAGEMENT*** ➜ ***REQUEST CERT*** is the same you are taken to when running the IPSec Wizard for an initial configuration. You can, therefore, make use of SCEP during IPSec Wizard configuration, too.

There are two points to observe when using SCEP. Both are due to the high degree of automation involved in an SCEP enrollment:

The enrollment task has to be specified in a single step, i.e. if you start it from the SNMP Shell, you need to specify all the necessary parameters in one line. When using the Setup Tool, make sure to enter all parameters correctly.

The creation of a new entry in the **CertMgmtTable** immediately schedules the enrollment task. If, e.g., you forget to specify the Subject Alternative Name and the enrollment succeeds before you can change the task specifications, you will receive a certificate without any Subject Alternative Name. In most cases there is no way to obtain another certificate with the desired parameters without previous interaction with the CA.

The Autosave function which is activated per default on your router is highly useful for ensuring the correct processing of a certificate enrollment even across a reboot of your router. In a single case, it can, however, happen that the configuration of your router is damaged. This happens when the router is powered off or rebooted while it is autosaving the configuration to save the state of a certificate enrollment.

Backing up configuration files is, therefore, highly recommended.

2.2 RADIUS Preset IPSec Peers

System Software Release 6.3.1 introduces an extension of our RADIUS (Remote Authentication Dial-In User Service) implementation that enables you to centrally store and maintain IPSec configurations on a RADIUS server. These configurations are retrieved from the server when the router boots and are temporarily stored in the respective MIB tables.

This has the advantage that if several routers use the same IPSec Peer List, this list need not be maintained on each of the routers. This reduces the efforts for synchronizing the router configuration once a change in the configuration has been made. An already existing RADIUS infrastructure (used for authentication, accounting or shell login) can be additionally used for this purpose.

Moreover, routers with large configuration files are available for other functions than IPSec immediately after the boot process has been completed. This means that, e.g., routing is available before all IPSec peers have been loaded which might take up to a couple of minutes if a large number of peers is statically stored on the router.

Configuration is necessary on both sides, the router and the RADIUS server: The router must be configured to query the RADIUS server and then retrieve the IPSec Peer list. The RADIUS server must have the necessary entries of the IPSec Peer list specified in the Users Database.

Router Configuration

Setting up the router to query the RADIUS server and retrieve the peer list is simple: The configuration is carried out in the menu ***IP*** ➧ ***RADIUS SERVER*** ➧ **ADD/EDIT** and is analogous to the configuration of a RADIUS server for other purposes.

In order to support the new functionality, the field **Protocol** can take the value *IPSEC configuration preload*. The entire rest of the configuration proceeds as is described in the chapter "RADIUS" of the Software Reference which is available for download from www.bintec.net.

We recommend securing traffic between the router and RADIUS server by IPSec. See "The Procedure", page 15, for details why this is highly recommended.

RADIUS Server Configuration

The list of IPSec peers is stored on the RADIUS server as a set of entries in the Users Database. Each user specification represents a single IPSec peer.

Note that you need a RADIUS dictionary extension in order to use the newly designed attributes. You can download this dictionary extension from www.bintec.net.

The actual syntax of the user specification highly depends on the RADIUS server used. Each peer specification, however, must include the following:

```
<user name> Password=<RADIUS default password>
<peer attribute>="<attribute value>"
<traffic attribute>="<attribute value>"
```

The syntax variables explain as follows:

Syntax Variable	Values
`user name`	This stands for *ipsecpre-<index>* with *<index>* being a decimal value including *0*.

Syntax Variable	Values
`RADIUS default password`	This has to be the value configured for the variable **DefaultPW** of the **radiusServerTable**.
`peer attribute`	This stands for *BinTec-ipsecPeerTable*. At least one peer attribute must be specified for a peer; if more than one peer attribute is specified, the variable assignments of all peer attributes are applied to the same peer entry in the MIB table of the router.

`(peer)`
`attribute value`
	The peer attribute value specifies a MIB variable from the **ipsecPeerTable** and its desired value in the following form:
	ipsecPeerPeerIds=peer-0.

`traffic attribute`	This stands for *BinTec-ipsecTrafficTable*. The peer specification can contain any number of traffic attributes, including none at all. The traffic attributes contained in a peer specification are linked in the sequence they are specified on the RADIUS server

`(traffic)`
`attribute value`
	The traffic value specifies a MIB variable from the **ipsecTrafficTable** and its desired value in the following form:
	*ipsecTrafficLocalAddress=192.168.1.0*.

Table 2-2: RADIUS server configuration syntax

The attribute values for the BinTec specific attributes consist of a string with variable assignments separated by spaces, e.g.:

```
ipsecPeerPeerIds=peer-0 ipsecPeerDescription=peer-0
```

This syntax is the same for both, the traffic and the peer attribute.

Note that the length of a single attribute is restricted to 253 characters. While peer entries can be distributed across several attributes, the traffic list entries have to be specified in a single one.

In order to save space, the table prefix of an attribute value can be omitted, e.g. *Local Address* instead of *ipsecTrLocalAddress*. Moreover, the "enum" variables can be specified instead of their description, e.g. *AuthMeth=1* instead *AuthMeth=pre_sh_key*.

When the peer list retrieved from the server is combined with the statically configured peers, read only variables are ignored, even if they have been specified by the Users Database on the RADIUS server. Moreover, the variables **ipsecPeerTrafficList**, **ipsecPeerNextIndex** and **ipsecTrafficNextIndex** are overwritten by the router when the two kinds of entries are combined.

The Procedure

Once both, router and RADIUS server, have been configured, the IPSec Peer List retrieval proceeds in four steps:

- At bootup all statically configured IPSec Peers are activated.
  If the IPSec Default Rule is set to *pass*, all traffic that is intended for protection by the peer entries stored on the RADIUS server is transmitted in plain text until the RADIUS peers have been activated. If you want to avoid this, you can set the IPSec Default Rule to drop. In this case you must make sure, however, that the RADIUS traffic necessary for retrieving the peer lists is allowed to pass. You can do this either by specifying a Pre IPSec Rule or by setting up a secure connection to the RADIUS server by statically configuring it as an IPSec peer (the RADIUS server must support this option). The second option is highly recommendable, since otherwise the Preshared Keys possibly included in the RADIUS peer entries are transmitted in plain text.

- After the router has completed the boot process, it queries the RADIUS server and requires the peer list. If no RADIUS server for IPSec peer list retrieval is available, loading immediately stops and only the statically configured peers are activated.

> The download of the peer list from the RADIUS server is initiated only if IPSec is activated. In order to be able to activate IPSec, at least a single peer has to be statically configured on the router.
>
> It is best to specify the RADIUS server as IPSec peer and retrieve the peer list securely.

- Each peer retrieved from the RADIUS server is installed on the router and queued before the statically configured peers.

> The peer list retrieved from the RADIUS server is not permanently stored on the router, nor is it saved with the rest of the configuration when using the save or export commands.

- When all preset peers have been retrieved from the RADIUS server, the peer list is activated. Doing so momentarily discards all existing IPSec tunnels.

2.3 DSL Trace

System Software Release 6.3.1 and BRICKware for Windows 6.3.1 support tracing DSL and other Ethernet connections. This is done in either of two ways:

- Using the **DIME Tools** included in **BRICKware for Windows**. The setup is similar to starting an ISDN trace. The **DIME Tools** come with an online help system you can refer to in case any questions about how to use this feature arise.
- In the SNMP shell by entering the `trace` command with the required arguments. By entering `trace -?`, help on using the command is displayed.

2.4 QoS in a LAN

Before System Software Release 6.3.1 it was not possible to extensively configure Quality of Service (QoS) in the Setup Tool for LAN interfaces. Even though it could be activated in the Setup Tool, the menus ***QoS*** ➜ ***Interfaces and Policies*** ➜**EDIT** ➜ ***QoS Scheduling and Shaping*** and ***QoS*** ➜ ***Interfaces and Policies*** ➜ **EDIT** ➜***Class-Based QoS Policies*** were not accessible for LAN interfaces, and configuration of all parameters included in these menus was possible only through using the SNMP shell.

In System Software Release 6.3.1, the parameters of both menus can be configured for LAN interfaces using the Setup Tool. You will find detailed information on the configuration options in the **QoS** manual which is available for download from www.bintec.net.

2.5 Software MPPC

Before System Software Release 6.3.1 MPPC (Microsoft Point to Point Compression) was only available through additional hardware (an expansion card carrying the respective chip was needed). MPPC is now available for all PPP WAN partners as an software option, too. You can activate MPPC in the menu ***WAN Partner*** ➜ **ADD/EDIT** by setting the field **Compression** to *MPPC*.

The availability of software MPPC significantly enhances the available PPP configuration options, since it is the only compression method that works together with MPPE encryption supported by all BinTec routers.

Note that for using software MPPC, you need the STAC license which is part of the Easy Licensing mechanism and thus already activated in the ex works state of the router. If, however, the Easy License has been deactivated, software STAC is not available.

2.6 MPPE with Hifn

System Software Release 6.3.1 introduces hardware supported MPPE encryption. If your router is equipped with a Hifn chip for hardware encryption (either on a XT-ENC or a XT-VPN resource module), any MPPE encryption is automatically performed by the Hifn chip and will not increase the load of the router CPU.

2.7 TCP MSS Clamping for Non-NAT Interfaces

Controlling the maximum size of an IP packet using PMTU (Path Maximum Transfer Unit) Discovery is not always successful. To obviate problems that might arise from the deficiencies of PMTU Discovery, TCP MSS (Maximum Segment Size) Clamping is used to control the size of TCP segments. So far this function has been available for NAT interfaces only; System Software Release 6.3.1 introduces TCP MSS clamping for Non-NAT interfaces.

TCP MSS clamping must be configured in the SNMP shell (**ipExtIfTable**). The variable **TcpMssClamping** can assume the following values:

Value	Meaning
0	MSS is set to automatic detection.
>0	Possible values are *1-32000*. The MSS is set to the value specified.
-1	MSS clamping is disabled.

Table 2-3: **TcpMssClamping**

2.8 Bridging for Frame Relay

System Software Release 6.3.1 introduces the possibility to enable bridging for Frame Relay interfaces. You can activate Bridging for a Frame Relay interface in the menu ***FR*** ➧ ***MULTIPROTOCOL OVER FRAME RELAY*** ➧ **ADD/EDIT** ➧ ***BRIDGE***.

Note that this feature is not available for PPP interfaces using a Frame Relay encapsulation.

2.9 X.25 to TCP Gateway

The X.25 to TCP Gateway is used to convert X.25 calls into TCP calls (also called TCP sessions). This requires that the specifics of the TCP and X.25 protocols are mapped to one another. In so far as the X.25 to TCP Gateway converts entire calls and not single datagrams, this feature differs from sending X.25 traffic over TCP networks and from sending TCP traffic over X.25 networks. For these tasks, XoT and Multiprotocol Routing over X.25 can be used.

The gateway cannot be configured using the Setup Tool, but needs to be configured in the SNMP shell. All relevant parameters are specified in the newly created **x25ToTcpTable**.

X.25 to TCP conversion

When the router receives an X.25 call, it checks the **x25ToTcpTable** for a matching entry. The decision whether to convert the X.25 call or not is based on any combination of the following parameters:

- **X25LocAddr** - the local address of the X.25 call
- **X25LocNSAP** – the Network Service Access Protocol used on the local side

- **X25RemAddr** – the destination address of the X.25 call
- **X25RemNSAP** – the Network Service Access Protocol used on the remote side
- **X25ProtocolID** – the protocol ID of the X.25 call
- **X25CallUserData** – the call user data of the X.25 call.

Only if the **x25ToTcpTable** contains a matching entry, a TCP call is established using the TCP call parameters specified in the respective entry. They are:

- **IpRemAddr** – the destination address of the TCP call
- **IpRemPort** – the destination port of the TCP call.

For converting X.25 calls into TCP calls, these parameters must be specified, since otherwise a TCP connection cannot be established.

After the outgoing TCP call has been accepted, the router accepts the incoming X.25 call, and data are transferred from one call to the other.

All calls that are received on the LOCAL interface (**ifIndex** *1*) are transferred to the X.25 to TCP Gateway for possible conversion. Therefore, all relevant X.25 calls need to be routed to that interface by X.25 routing. If any other X.25 calls are accepted on the local interface, too, the client that answers first receives the call.

The system usually accepts PAD calls on the local interface. If this behavior is undesired, it can be disabled using the command `LocalPadCall=dont_accept`.

TCP to X.25 conversion

The X.25 to TCP Gateway does not only convert X.25 calls into TCP calls, but allows to convert TCP calls into X.25 calls, also. The necessary parameters are specified in the same MIB table (**x25toTcpTable**).

The **x25ToTcpTable** must specify a local IP port (**IpLocPort**) on which to listen for TCP calls that must be converted. As soon as a TCP call is received on this port, the router checks the **x25ToTcpTable** for any combination of the following parameters:

- **IpLocAddr** – the local IP address of the TCP call
- **IppRemAddr** – the remote IP address of the TCP call
- **IpRemMask** – the netmask of the remote IP address
- **IPRemPort** – the remote port the call must be established to
- **IPRemPortRange** – the range of remote ports specified for the TCP call.

If a matching entry is found, an X.25 call with the following parameters is established:

- **X25LocAddr** - the local address of the X.25 call
- **X25ProtocolID** – the protocol ID of the X.25 call
- **X25CallUserData** – the call user data of the X.25 call.

After the outgoing X.25 call has been accepted, the router accepts the incoming TCP connection, too, and data are transferred from one call to the other.

When either of the calls is cleared, the other call is cleared, too. The clear parameters are lost in the process, since X.25 clear parameters cannot be fitted into a TCP disconnect.

Packetizing

Normally, the TCP packet information is lost when the TCP call is converted into an X.25 call. If the TCP application, however, needs to know the structure of packets created from an X.25 call or, respectively, needs to shape the packets sent to the X.25 call, packetizing can be enabled for the TCP call. In this case a header is added specifying the packet size.

Configuration Example

An entry into the **x25ToTcpTable** is created by specifying all desired parameters in a single command, e.g.:

```
MyRouter:> x25totcp

inx Index(*rw)             Descr(rw)            State(-rw)
    X25LocAddr(rw)         X25LocNSAP(rw)       X25RemAddr(rw)
    X25RemNSAP(rw)         X25ProtocolId(rw)    X25CallUserData(rw)
    IpLocAddr(rw)          IpLocPort(rw)        IpRemAddr(rw)
    IpRemMask(rw)          IpRemPort(rw)        IpRemPortRange(rw)
    Metric(rw)             Direction(rw)        Packetizing(rw)
    Reset(rw)              Intr(rw)             Monitor(rw)
    MonState(ro)

MyRouter:x25ToTcpTable>Index=0 Direction=x2t x25CallUserData="POS9"
ipRemAddr=1.2.3.4 ipRemPort=4711 Packetizing=atos
```

According to this entry all X.25 calls with their X.25 Call User Data field starting with the ASCII characters "P", "O", "S" and "9" will be sent to IP address 1.2.3.4 and port 4711. The X.25 local address is irrelevant for mapping the packet specifics as well as the X.25 Protocol ID. The IP packets are generated according to the Atos POS specification.

MIB Description

The variables you can find in the **x25ToTcpTable** have the following relevance:

Variable	Meaning
Index	The unique number of the MIB entry.
Descr	This variable specifies a description of the entry. You can use arbitrary text here.

Variable	Meaning
State	The state of the entry. Possible values are: - *valid* : The entry is valid and used by the system. - *invalid*: The entry is not used/disabled. - *delete*: The entry will be deleted.
X25LocAddr	The local X.121 address of the X.25 call. If not specified, the local address will not be considered for matching the protocol specifics. Extended addresses are indicated by a leading "@". The wildcards "*", "?", "[", "]", "{", "}" may be used.
X25LocNSAP	The local NSAP (Network Service Access Protocol) of the X.25 call. If not specified, the local NSAP will not be considered for matching the protocol specifics. The NSAP is preceded by an "X" if it is an OSI compatible NSAP or by a "N" if the NSAP is in a not OSI compatible format. The wildcards "*", "?", "[", "]", "{", "}" may be used.
X25RemAddr	The remote X.121 address of the X.25 call. If not specified, the remote address will not be considered for matching the protocol specifics. Extended addresses are indicated by a leading "@". The wildcards "*", "?", "[", "]", "{", "}" may be used.

Variable	Meaning
X25RemNSAP	The remote NSAP (Network Service Access Protocol) of the X.25 call. If not specified, the remote NSAP will not be considered for matching the protocol specifics. The NSAP is preceded by an "X" if it is an OSI compatible NSAP or by a "N" in the NSAP is in a not OSI compatible format. The wildcards ’*’, ’?’, ’[’, ’]’, ’{’, ’}’ may be used.
X25ProtocolId	The protocol ID of the X.25 call. If this variable is set to *-1*, it will be used as a wildcard, and any protocol ID is accepted.
X25CallUserData	The call user data field of the X.25 call packet following the protocol ID. If not specified, the call user data field is not considered for matching the protocol specifics.
IpLocAddr	This variable specifies the Local Address of the IP datagrams. If the local address is not specified (i.e. set to *0.0.0.0*), this variable is not considered for matching the protocol specifics.
IpLocPort	This variable specifies the local IP port number. If this variables is set to *-1*, the local port number is not specified and is not considered for matching the protocol specifics.
IpRemAddr	This variable and the variable **x25ToTcpIpRemMask** together specify the range of the remote IP addresses. If this variable and **x25ToTcpIpRemMask** are both set to *0.0.0.0*, the remote IP address is not considered for matching the protocol specifics.

Variable	Meaning
IpRemMask	This variable and the variable **x25ToTcpIpRemAddr** together specify the range of the remote IP addresses.
IpRemPort	This variable and the variable **x25ToTcpIpRemPortRange** together specify the range of remote IP port numbers. If this variables is set to *-1*, the remote port number is not specified and is not considered for matching the protocol specifics.
IpRemPortRange	This variable and the variable **x25ToTcpIpRemPort** together specify the range of remote IP port numbers. All port numbers between and including the values specified for these two variables are considered to be within the range.
Metric	The metric assigned to an entry specifies the order in which the rules for mapping X.25 and TCP packets are to be applied. This is important for backup and redundance scenarios. The lower the value, the higher is the importance of the entry.
Direction	This variable specifies whether the entry it belongs to is valid in direction from TCP to X.25 (*t2x*), X.25 to TCP (*x2t*) or in both directions (*both*). The parameters used for matching the specifics of the destination protocol should not contain wildcards.

Variable	Meaning
Packetizing	This variable specifies, how datagrams are encoded in the TCP stream: ■ *none*: No packetizing, packetizing information is lost. ■ *atos*: Packetizing according to the Atos specification for POS (Point of Sales) terminals. ■ *rfc1006*: Packetizing according to RFC 1006.
Reset	This variable specifies, how the reception of a Reset Packet on the X.25 link is handled: ■ *clear*: The connection is cleared. ■ *accept*: The reset is confirmed, and data transfer continues.
Intr	This variable specifies, how the reception of an Interrupt Packet on the X.25 link is: ■ *clear*: The connection is cleared. ■ *ignore*: The interrupt is ignored, and data transfer continues. ■ *pass*: The data part is sent to TCP like normal data, and data transfer continues. Note that the interrupt data cannot be distinguished from normal data in the TCP stream.

Variable	Meaning
Monitor	This variable applies to entries with **Direction** set to *x2t* (X.25 to TCP) only. In a backup or redundance scenario, the remote IP address has to be monitored in order to decide if a backup entry has to be used. If **Monitor** is set to *on*, the current entry in the **x.25ToTcpTable** is valid only if **IpRemoteAddr** replies to ICMP Echo Requests. Otherwise this entry is considered invalid, and an alternate entry aiming at a different IP address is used.
MonState	This variable displays the reachability of the Remote IP Address: ■ *not_monitored*: The address is not monitored. ■ *unreachable*: The address is monitored, but does not reply to ICMP Echo Requests. ■ *ready*: The address is answering ICMP Echo Requests and is ready to accept TCP calls.

Table 2-4: **x25ToTcpTable**

2.10 `show rev`

Using the newly created SNMP command `show rev` displays the version of the major software components (Logic, Bootmonitor and System Software (Boss)) at a single glance without the need of rebooting the router. Before System Software Release 6.3.1 the version of the Boot Monitor was inaccessible without a reboot.

3 Changes

In addition to new features introduced with System Software Release 6.3.1, there has been a number of changes enhancing the functionality of your router:

- Functions Available in X1000/X1200/X3200 (chapter 3.1, page 28)
- RADIUS Callback (chapter 3.2, page 29)
- DNS Improvement (chapter 3.3, page 29)
- Keepalive Monitoring (chapter 3.4, page 29)
- Leased Line License (chapter 3.5, page 30)
- PPTP Improvements (chapter 3.6, page 30)
- MPPE V2 RFC 3078 (chapter 3.7, page 30)
- Delayed SA Deletion in IPSec (chapter 3.8, page 31)
- SIF – Configurable Time-Out (chapter 3.9, page 31)
- Initial IP Configuration (chapter 3.10, page 32)

3.1 Functions Available in X1000/X1200/X3200

Due to the extensive new features offered by System Software Release 6.3.1, we had to remove the X.25 subsystem from the IPSec version of System Software Release 6.3.1 for the following Routers:

- **X1000**
- **X1200**
- **X3200**

Note that this restriction applies only to routers running the IPSec version of System Software Release 6.3.1.

3.2 RADIUS Callback

When an initial call to a router is authenticated via a RADIUS server, and if callback is negotiated between the calling host and the router, then the authentication of the callback is performed via the RADIUS server, too. For this purpose, a second authentication request is sent to the RADIUS server.

If the RADIUS server, however, is configured to evaluate the Calling Station ID and the Called Station ID, a problem can arise: From the perspective of the router, the Called Station ID of the first call is that the router itself, the Calling Station ID is that of the calling host. The Called station ID of the second call (again from the perspective of the router that now performs the callback) is that of the host, and the Calling station ID is that of the router.

In order to avoid the necessity of additional entries in the Users Database of the RADIUS server, BinTec has designed a mechanism by which both IDs can be transmitted in both authentication requests (for the initial call as well as for the callback call) in such a way that they can be correctly evaluated and no additional configuration of the RADIUS server is necessary.

3.3 DNS Improvement

System Software Release 6.3.1 enhances the DNS mechanism of BinTec routers, so that compressed names and great numbers of DNS requests can be easily handled.

Moreover, this solves a problem with hostname resolution in the Setup Tool and a minor memory leak.

3.4 Keepalive Monitoring

The number of groups that can be configured for Keepalive Monitoring in the menu ***SYSTEM*** ➧ ***KEEPALIVE MONITORING*** has been increased from ten to 256.

When configuring large number of hosts to monitor, keep in mind that the amount of RAM required will increase, too.

3.5 Leased Line License

The license mechanism of System Software Release 6.3.1 has been adjusted so that the following routers from the family of **X2000** routers are able to handle BRI leased lines without the need for a license:

- **X2100**
- **X2402**
- **X2404**.

Note that if you have a leased line license installed on one of these routers, the license will show as *not_supported* or *invalid_license* in the ***LICENSES*** menu. You can safely delete these licenses from your router if you are using System Software Release 6.3.1. Remember to reinstall them in case you return to an earlier version of our system software.

3.6 PPTP Improvements

BinTec's PPTP implementation has been improved to be fully RFC 2637 compliant. This also solves a number of problems that have been verified with earlier versions of our System Software.

3.7 MPPE V2 RFC 3078

System Software Release 6.3.1 introduces a new version of MPPE V2 which is fully compliant with RFC 3078. This ensures interoperability with Windows 2000 and XP hosts and with MS-CHAP V1 and V2 authentication.

MPPE V2 (RFC 3078) does not replace MPPE V2 as it has been available before, but is available as an additional value for the field **Encryption** in the ***WAN PARTNER*** ➧ **ADD/EDIT** menu.

3.8 Delayed SA Deletion in IPSec

With PFS (Perfect Forward Secrecy) enabled for a specific peer in an IPSec configuration, the following problem could arise: When the last packet of an IKE Phase 2 negotiation happened to be lost, the initiator immediately deleted the IKE Phase 1 SA and sent the Delete Notification to the responder. The responder then cancelled Phase 2 negotiation, and no tunnel was established.

System Software Release 6.3.1 introduces a configurable delay for Phase 1 SA deletion: The variable **ipsecGlobContPfsIdentityDelay** specifies how many seconds the router will wait before finally deleting the Phase 1 SA. The default value is eight seconds which (using the default configuration for retries and time-outs) covers four retries to continue Phase 2 SA negotiation.

In general it will not be necessary to change the value of **ipsecGlobContPfsIdentityDelay**, but if the connection is known to be bad or if the timing parameters have been changed, we suggest adjusting this value, too.

3.9 SIF – Configurable Time-Out

Before System Software Release 6.3.1 it was not possible to specify the time-out used for TCP/UDP sessions. It now is possible to configure the time-out of TCP, UDP and PPTP sessions according to your needs by assigning the desired value to any of the variables **ipSifUdpTimeout**, **ipSifTcpTimeout** or **ipSifPPTPTimeout** in the **ipSifTable**. Configuration needs to be made in the SNMP shell, the range of possibe values is from *30* to *86400* seconds.

The default values are:

- for UDP sessions *180* seconds

- for TCP sessions: *3600* seconds
- for PPTP sessions (TCP port 1723 and GRE): *86400* (24 hrs.).

3.10 Initial IP Configuration

Upon first booting a BinTec router, the administrator had to provide an IP address and the corresponding netmask either via a serial connection or via BOOTP. While both methods are still available, routers now have a preconfigured IP address: It is set to 192.168.0.254 and the netmask is accordingly set to 255.255.255.0.

The router will stop broadcasting BOOTP requests when either a new IP address is entered or the first TCP session is accepted using the IP address provided.

4 Bugfixes

The following problems have been solved in System Software Release 6.3.1:

- SNMP Shell – Zero Size Requests (chapter 4.1, page 34)
- X.25 with **X8500** (chapter 4.2, page 35)
- IPSec
  - Configuration of Phase 2 Proposals (chapter 4.3.1, page 35)
  - Error in ***ADVANCED SETTINGS*** Menu Display (chapter 4.3.2, page 35)
  - IPSec Menu Visible without IPSec License (chapter 4.3.3, page 36)
  - IPSec Panic with Missing Peer ID (chapter 4.3.4, page 36)
  - Setup Tool Crash During Configuration (chapter 4.3.5, page 36)
  - Stacktrace in Setup Tool (chapter 4.3.6, page 37)
  - False IP Address Used for SA Creation (chapter 4.3.7, page 37)
  - Settings for Traffic List Entry not Saved (chapter 4.3.8, page 38)
  - IPSec Wizard – CRL Distribution Point not Considered (chapter 4.3.9, page 38)
  - Inoperative IPSec SAs (chapter 4.3.10, page 38)
  - Import of PKCS #7 Certificates not Possible in Setup Tool (chapter 4.3.11, page 39)
  - Index Value not Changed after Setup Tool Configuration (chapter 4.3.12, page 39)
- QoS
  - **Transmit Rate** Set to *0* (chapter 4.4.1, page 39)
  - Bound Transmit Rate in Setup Tool (chapter 4.4.2, page 40)
- SIF (Stateful Inspection Firewall)
  - Locally Initiated TCP Sessions Time Out (chapter 4.5.1, page 40)
  - Setup Locks (chapter 4.5.2, page 41)
- DynVPN – Router Lock Up (chapter 4.6, page 41)

4.1 SNMP Shell – Zero Size Requests

When receiving zero size packets, the SNMP shell tried to process them as normal packets. Since there is no procedure for such packets, the SNMP shell locked up.

This Problem has been solved: The SNMP shell now discards zero size packets.

4.2 X.25 with X8500

On **X8500**, there were problems with X.25 connections when running System Software 6.2.2. These problems appeared, e.g., as sporadic reboots.

This problem has been solved: **X8500** now handles X.25 correctly.

4.3 IPSec

A number of problems have been solved in the range of IPSec functions:

4.3.1 Configuration of Phase 2 Proposals

The menu ***IPSec*** ➔ ***Configure Peers*** ➔ **APPEND/EDIT** ➔ ***Special Settings*** ➔ ***Phase 2*** did not allow to specify and save exactly that Phase 2 proposal which corresponded to the proposal configured as Phase 2 default. Upon saving the value was reset to the default (marked by the suffix "(def)"). If the default was subsequently changed, the setting made for the peer specifically changed, too. This could result in problems if the peer did not support the changed proposal. Configuring the Phase 2 proposal in the SNMP shell was possible, though.

This problem has been solved.

4.3.2 Error in Advanced Settings Menu Display

Upon entering the menu ***IPSec*** ➔ ***Advanced Settings***, the field **Cookies Size** was displayed, even if the **Use Zero Cookies** field was set to *no*.

This was merely a display problem and did not affect the functionality of our routers. If **Use Zero Cookies** was set to *no*, the **Cookies Size** field disappeared as soon as you put the cursor on it, and if **Use Zero Cookies** was set to *yes*, **Cookies Size** became configurable. The problem has been solved.

4.3.3 IPSec Menu Visible without IPSec License

If a router was equipped with an IPSec software image, but was missing an IPSec license, the ***IPSEC*** menu was visible and could be accessed. If an IPSec configuration was saved in this state, all IP traffic was blocked, and the router was not accessible via Telnet.

This situation could arise, too, if you deleted the IPSec license from a router that already had an IPSec configuration.

This problem has been solved: The IPSec menu is accessible only if the relevant license is installed on the router.

4.3.4 IPSec Panic with Missing Peer ID

If during IPSec parameter negotiation the remote peer's ID had not been transmitted as soon as Phase 1 negotiation was complete, the router rebooted.

This problem was due to an error within the Certificate Lookup function. It occurred only if there was no peer with dynamic IP address configured on the router, and if no ID was configured for the peer the tunnel was to be set up to. The problem could be obviated by creating a dummy dynamic peer with the desired Phase 1 parameters, but without any traffic list entries.

This problem has been solved.

4.3.5 Setup Tool Crash During Configuration

When configuring either a CA Certificate or an Own Certificate in the Setup Tool, the Setup Tool occasionally crashed. This happened only if the certificate contained characters that appeared as "printf" format descriptors.

This problem has been solved: Characters appearing as "printf" descriptors are simply displayed and not interpreted.

4.3.6 Stacktrace in Setup Tool

With system software 6.2.5 IPSec, it could happen that the Setup Tool crashed when either of these two IPSec submenus was accessed:

- ***IPSec*** ➧ ***IKE (Phase 1) Defaults***
- ***IPSec*** ➧ ***Configure Peers*** ➧ **EDIT** ➧ ***Special Settings*** ➧ ***Phase 1***.

Another stacktrace occurred when

- entering the menu ***IPSec*** ➧ ***Monitoring*** ➧ ***IKE Securty Associations***
- displaying the detail of any IKE SA
- returning to the previous menu and
- hitting **cursor up** a few times.

Occasionally, this caused a reboot of the router, also. Both problems have been solved.

4.3.7 False IP Address Used for SA Creation

An IPSec tunnel established between two peers by IPSec Callback was occasionally inoperative if there was no **Peer Address** or Fully Qualified Domain Name (FQDN) configured for the remote peer that requested the callback. Occasionally, the same error occurred without IPSec callback if the central side initiated rekeying.

This problem was due to the fact that, in this context, the local router chose the last IP address contained in the **ipAddrTable** for SA negotiation. This IP address, however, was not necessarily the correct one: If the IP address chosen was not the same as the source address contained in the data packets, non functional Phase 2 SAs were negotiated. The problem has been solved.

4.3.8 Settings for Traffic List Entry not Saved

When creating a new traffic list entry, the menu ***IPSec*** ➧ ***Configure Peers*** ➧ **EDIT** ➧ **APPEND** (traffic list entry) ➧ ***Special Settings*** for this entry was accessible, even if the entry had not yet been saved in the MIB tables. This had the effect that any changes made in the ***Special Settings*** menu could not be saved.

This problem has been solved: If any changes are made to an traffic list entry which has not been written to the MIB, theses settings are remembered and applied once the entry is saved in the MIB.

4.3.9 IPSec Wizard – CRL Distribution Point not Considered

In case a CA certificate did not contain a CRL distribution point and no certificate server was configured either, the IPSec Wizard started every time the IPSec main menu was accessed. This was the case even if the Own Certificate or a Peer Certificate contained a CRL distribution point.

This problem was due to the fact that other certificates than the CA certificate were not taken into account when checking for a CRL distribution point. It has been solved: All certificates are now checked.

4.3.10 Inoperative IPSec SAs

Proposals specifying no encryption (**Encryption** *none*) could lead to inoperative IPSec SAs (Security Associations) when the router was equipped with a Hifn module.

This problem has been solved.

4.3.11 Import of PKCS #7 Certificates not Possible in Setup Tool

Importing PKCS #7 certificates was not possible with the Setup Tool. If you needed to import a PKCS #7 certificate, you had to use the "cert" application from the SNMP shell.

This problem has been solved: PKCS #7 certificates can be imported using the Setup Tool.

4.3.12 Index Value not Changed after Setup Tool Configuration

When configuring IPSec without having run the IPSec Wizard before, the Pre IPSec Rules entered were not saved on the router. This could result in an inoperative configuration when IKE traffic could not be passed in plain text.

This problem has been solved.

4.4 QoS

A number of problems have been solved in the range of QoS functions:

4.4.1 Transmit Rate Set to *0*

When entering a new QoS policy in the Setup Tool (***QoS*** ➜ ***Interfaces and Policies*** ➜ **EDIT** ➜ ***Class-Based QoS Policies*** ➜ **ADD**) and not specifying a **Transmit Rate**, the actual setting created in the MIB tables was **TxRateLimitation** *bounded* and **TxRate** *0*. This had the effect that no packet from this queue was actually transmitted.

This problem occurred only if you had created a high priority queue before, and if you had not saved the configuration before creating the new policy. It has been solved.

4.4.2 Bound Transmit Rate in Setup Tool

In rare cases, even though you could change the value of the field **Bound Transmit Rate** in the menu ***QoS*** ➜ ***Interfaces and Policies*** ➜ **EDIT** ➜ ***Class-Based QoS Policies*** ➜ **ADD/EDIT**, the change was not effective. Upon reentering the menu, it was reset to *no*. It was, however, possible to configure the value using the SNMP shell.

Moreover, when setting the field **Transmit Rate** to a value of *0*, the value for the variable **TxRateLimitation** in the **qosPolicyTable** should have been reset to *not_bounded*. This, however, did not work.

This problem has been solved: All changes are executed and saved properly.

4.5 SIF (Stateful Inspection Firewall)

A number of problems have been solved in the range of SIF functions:

4.5.1 Locally Initiated TCP Sessions Time Out

If a TCP session was initiated on the router itself (e.g. if a Telnet session was started), this session timed out after only 30 seconds of inactivity. The time-out was not configurable, either.

This problem was due to an error in the SIF software. It has been solved.

4.5.2 Setup Locks

If one or more entries were deleted from the SIF filter configuration using the Setup Tool, and the Stateful Inspection menu was then left with **SAVE**, the Setup Tool locks up.

This problem was due to an error in the list handling. It has been solved.

4.6 DynVPN – Router Lock Up

After a maximum of 128 instances of trying to propagate or propagating the dynamic IP address of a router in a DynVPN via PPTP configuration, the router did not process any further IP address updates and eventually locked up.

This problem has been solved.

4.7 CAPI – Problems with High CAPI Traffic Volume

When disconnecting more than 20 CAPI sessions at once, it could happen that the router locked up. Moreover, more than 120 CAPI sessions were not supported.

These problems have been solved, and the CAPI implementation of System Software Release 6.3.1 properly supports up to 254 CAPI sessions.

4.8 `modem status` Command

Entering the command `modem status <index>` in the SNMP shell led to an exception stacktrace without subsequent reboot of the router.

This problem has been solved.

4.9 `ping` Command

Entering the command `ping -?` returned an `unknown parameter` error message, and entering `ping` without any arguments returned an `invalid address (null)` error message.

Moreover, extensive usage of the `ping` command could lead to a minor memory leak.

These problems have been solved: `ping -?` properly displays the help text without any confusing error messages, and entering `ping` without any arguments returns a `no args to work` message. Extensive use of the `ping` command does no longer lead to a memory leak.

4.10 XoT – Setup Tool

In the following menus it was not possible to select an interface for XoT configuration:

- ***X.25*** ➧ ***LINK CONFIGURATION***
- ***X.25*** ➧ ***ROUTING*** ➧ **ADD/EDIT**.

This problem has been solved: Interface selection is now possible.

4.11 Frame Relay – Interface State Remains Down

On a router with a serial interface configured as DTE for Frame Relay, it could happen that the interface state remained *down* after a link connection failure, even if layer 1 was restored.

This problem has been solved: As soon as layer 1 is restored, the interface state is set to *up* again.

4.12 configd – `get` with XMODEM

A file transfer from one router to another failed under the following circumstances:

- the `configd get` command was used via telnet in raw/binary mode (`telnet -rb`)
- the file transfer should be via XMODEM (the full syntax is displayed when entering `configd -?` in the SNMP shell).

The problem was due to an incorrect test in the Telnet client. It has been solved, and file transfer via Telnet/XMODEM works properly.

4.13 TFTP – Session Fails

A TFTP session with a PC running SuSE LINUX 8.1 failed with an #4 error.

This problem was due to corrupted data in the TFTP requests. It has been solved.

4.14 X.25 – Reboot with Ethernet Interface

If an Ethernet interface was chosen in a X.25 configuration, some BinTec routers crashed when a X.25 connection should be established.

This problem was due to an error within the Ethernet driver. It has been solved: Ethernet interfaces can now be used for X.25 on all our routers supporting X.25.

4.15 Syslog – Reboot

After printing a large number (several hundred) syslog messages to the serial console, the router occasionally rebooted.

This problem was due to a buffer overflow. It has been solved, and any number of syslog messages can be printed to the serial console.

4.16 DHCP-Relay Inoperative

When **X1200** was used to relay DHCP messages, the DHCPNACK message coming from the DHCP server was not relayed to the PC requesting an IP address.

This problem has been solved: All BinTec routers can be used for DHCP Relay without problems.

4.17 DNS Proxy – Load Problem

After a large number of DNS requests had been processed, the DNS Proxy no longer added any entries to the **ipDnsTable**, i.e. neither positive nor negative responses were cached.

This problem has been solved: Any number of resolved DNS requests can be processed.

4.18 X.21 – Reboot Needed After Configuration

After using the Setup Tool for initial configuration of a serial interface for X.21, the router needed to be rebooted for the configuration to become operative.

This problem has been solved: Once you save the configuration made in the Setup Tool it becomes immediately effective.

4.19 IP Accounting – Display Error

With IP Accounting activated, only the count of sent data packets, but not the count of received data packets was displayed.

This problem has been solved: IP packets are counted and displayed properly.

4.20 PPPoE – Reboot with STAC Compression

When a PPPoE interface was configured to use STAC compression, either hardware supported or not, the router occasionally rebooted.

This problem was due to an error within the PPPoE driver. It has been solved: STAC compression can be used on PPPoE interfaces without any problems.

4.21 X4000 - Reboot Loop with PRI Card

When a router of the **X4000** family was booted while a PRI cable was plugged into a PRI expansion card (X4E-1/2PRI) which was equipped with a resource module carrying a Hifn chip, the router occasionally entered a reboot loop.

This problem has been solved.

4.22 Saving and Restoring Configurations

When using a TFTP transfer (command `get` or `put`) for saving and restoring configuration files, a number of problems could occur:

- If the Ethernet interface was configured for any but the default setting (*auto*), and if the configuration file was then stored on a TFTP server, the default settings were still applied when this configuration was downloaded again: The Ethernet interface was not set, e.g., to *10 MBit Half Duplex* but to *auto*.
- When downloading a corrupted configuration file to the Flash Card of a **X8500**, it could happen under specific circumstances that all files were deleted from the Flash Memory.
- The router sent an configuration file with an empty **isdnStkTable** if the autoconfiguration of the ISDN interface was currently activated in the RAM of the router (**Autoconfig** in the **isdnIfTable** set to *on* or **ISDN Switch Type** set to *autodetect on bootup* in the ***WAN*** menu.).

All of these problems have been solved. The `get` and `put` commands now function properly.

5 Known Issues

Even though we thoroughly test our system software, the possibility of problems arising during every day use cannot be completely eliminated. BinTec has, therefore, created a mailing list (**release-info**) which will keep you informed on problems, solutions and workarounds verified in our laboratories. If you want to subscribe to this mailing list, you will find a respective link on the download pages of www.bintec.net.