



User Guide Addendum Release 4.3 Maintenance March 10, 2004

Introduction

This User's Guide Addendum provides information and procedures that will enable system administrators to configure and use the specific features introduced in the 4.3 Maintenance Releases for the Nomadix Universal Subscriber Gateway II™ (USG II).

The features covered are

- L2TP Tunneling
- Local Syslog and Syslog Filters
- RADIUS Proxy Accounting Logs
- IPSec Tunneling
- Remote Billing Mirror Record Change

Feature Configurations

L2TP Tunneling

Define Radius Service Profiles.

Radius service profiles are used to direct username access requests for both plain Radius users and users who supply realm/domain in their username. Steel-belted Radius servers can only be set to interpret tunnel profiles in either prefix or suffix-mode so a minimum of two Steel-belted Radius servers are required if both prefix and suffix-based usernames are to be handled. What these Radius servers will return in response to a Radius access request is the L2TP tunnel parameters that the USG II will use to establish an L2TP tunnel. See Figure 1 for an example of a Radius service profile.

- Create a Radius service profile to a Radius server that will handle Prefix-based users. This is to handle users that will login with a username in the format type of "ISP/username". In this case the delimiter is "/" and what appears before it, "ISP", is the realm name.
- Create a Radius service profile for a Radius server that will handle Suffix-based users. This is to handle users that will login with a username in the format type of

“username@ISP.com”. In this case the delimiter is “@” and what appears after it, “ISP.com”, is the realm name.

Unique Name:

Authentication

Enable RADIUS Authentication Service

Primary IP: Port: Secret Key:
Secondary IP: Port: Secret Key:

Accounting

Enable RADIUS Accounting Service

Primary IP: Port: Secret Key:
Secondary IP: Port: Secret Key:

Retransmission Options

Retransmission Method: Failover Round-Robin
Retransmission Frequency: (seconds)
Retransmission Attempts: (per server)

• Figure 1

Define Tunnel Profiles.

Tunnel profiles can be defined when L2TP tunnel parameters are known and it is not necessary to send an access request to a Radius server to obtain those parameters or for accounting purposes.

- Create a tunnel profile for each L2TP tunnel whose parameters are known. The tunnel parameters that the profile contains are the IP address of the LNS and the tunnel password. See Figure 2 for an example of a tunnel profile.

Unique Name:

Tunnel Parameters

Tunnel Peer IP:
Tunnel Password:

• Figure 2

Define Realm Routing Policies.

Realm routing policies are used to determine how supplied username/password input is used to authenticate users.

- Create a realm routing policy for each realm that will be handled. The realm routing policy will reference either a Radius service profile or a tunnel profile. Many different realm routing policies can reference the same Radius service or tunnel profile.

See Figure 3 for a realm routing policy that handles prefix-based usernames using a Radius service profile. Notice that “Specific Realm” is clicked and the “Realm name” is “cisp”. Also notice that “Prefix match only” is clicked and that the delimiter is “/”. This means that this realm routing policy will match usernames that are of the format “cisp/username”.

This policy references a Radius service profile so a realm match will result in an access request being sent to the Radius server(s) specified in the Radius service profile. In this case, the Radius service profile “RadiusPrefix” is referenced and so the Radius server(s) defined therein will receive Radius access requests.

Notice that the checkbox is unchecked for “Strip off routing information when sending to RADIUS server”. This box must always be unchecked in order to pass realm information to the Radius server(s) for matching of realm information to its defined tunnel profiles, which contain the needed tunnel parameters.

The checkbox “Strip off routing information when sending to tunnel server” may or may not be checked depending on the configuration of the tunnel server and how it will be authenticating subscribers. In this example, it is checked and so realm information will be stripped leaving only the simple username and password to be passed to the tunnel server.

The tunnel server in this case is configured to authenticate users via another Radius server that handles a single realm. Since it handles a single realm, no realm information is needed for users and so must be stripped. In this case, it is stripped by the USG II, but it could easily have been stripped by the tunnel server, or by the tunnel server’s Radius server. This is by design and for maximum flexibility.

Also note that the “Local hostname” field is blank which means that the USG II’s default local hostname of “usg_lac” will be used by the USG II. This allows for setting the local hostname to any desired value other than the default. The L2TP peers exchange their local hostnames during tunnel negotiation.

Specific Realm	<input checked="" type="radio"/>	Realm name:	<input type="text" value="cisp"/>
Wildcard match	<input type="radio"/>		

Prefix match only	<input checked="" type="radio"/>	(Match characters preceding "/")
Suffix match only	<input type="radio"/>	(Match characters following "@", i.e., NAI realm)
Match either	<input type="radio"/>	(Try prefix first, then try suffix if no prefix match)

RADIUS Service Profile:

Strip off routing information when sending to RADIUS server

Tunnel Profile:

<u>Tunnel Parameters (for profile-triggerred or RADIUS-triggerred tunnels):</u>	
Strip off routing information when sending to tunnel server	<input checked="" type="checkbox"/>
Local hostname:	<input type="text"/>

• Figure 3

See Figure 4 for a realm routing policy that handles suffix-based usernames using a tunnel profile. This differences in this example are that the realm name is “tcisp.com”, “Suffix match only” is enabled (the delimiter in this case is “@”), and a tunnel profile, “LNS-One”, is selected instead of a Radius service profile.

This means that this realm routing policy will match usernames that are of the format “username@tcisp.com”. Since this policy references a tunnel profile, no Radius access requests will be sent to any Radius server. In this case, the USG II will use the L2TP tunnel parameters specified in the tunnel profile to establish a tunnel and pass the username/password input to the tunnel server.

Again, as before, the username passed to the tunnel server will have realm information stripped since the checkbox for “Strip off routing information when sending to tunnel server” is checked. This checkbox may be unchecked if it is necessary for usernames to contain realm information for user authentication.

The “Local hostname” field is also blank in this example which means that the USG II will use the default value of “usg_lac” during tunnel negotiation.

Specific Realm Realm name:
 Wildcard match

Prefix match only (Match characters preceding "/")
 Suffix match only (Match characters following "@", i.e., NAI realm)
 Match either (Try prefix first, then try suffix if no prefix match)

RADIUS Service Profile: (select one) ▼

Strip off routing information when sending to RADIUS server

Tunnel Profile: LNS-One ▼

Tunnel Parameters (for profile-triggerred or RADIUS-triggerred tunnels):

Strip off routing information when sending to tunnel server

Local hostname:

Figure 4

Configure Radius Client.

The USG II Radius client must be setup for realm-based routing mode since realm information will be used by the USG II's L2TP tunnel feature to determine how to handle usernames that contain realm information. See Figure 5 for an example of setting the routing mode to handle realm-based usernames.

Server Selection

Routing Mode: Disabled Realm-Based Fixed

Default RADIUS Service Profile: NMDXRadius ▼

Figure 5

That should cover the main points regarding configuring an USG II to support L2TP tunneling. See below for screen snapshots of Steel-belted Radius server configuration settings required to support realm-based tunnel profile matching and tunnel parameter lookup.

Steel-belted Radius server configuration settings.

Figure A is an example of Steel-belted Radius configured to handle prefix-based realm usernames.

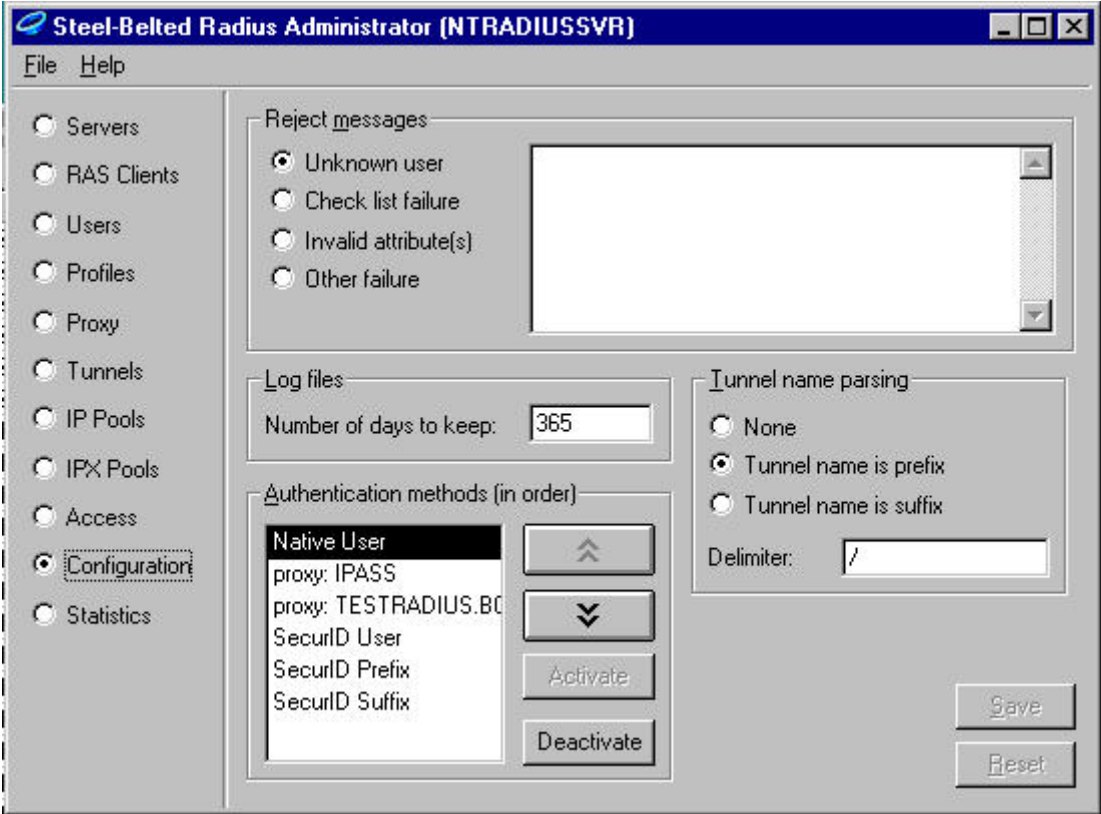


Figure A

Notice that “Tunnel name is prefix” is enabled for tunnel name parsing and that the delimiter is “/”. The delimiter must be “/” as that is only acceptable value on the USG II as a prefix delimiter.

Figure B is an example of Steel-belted Radius configured to handle suffix-based realm usernames.

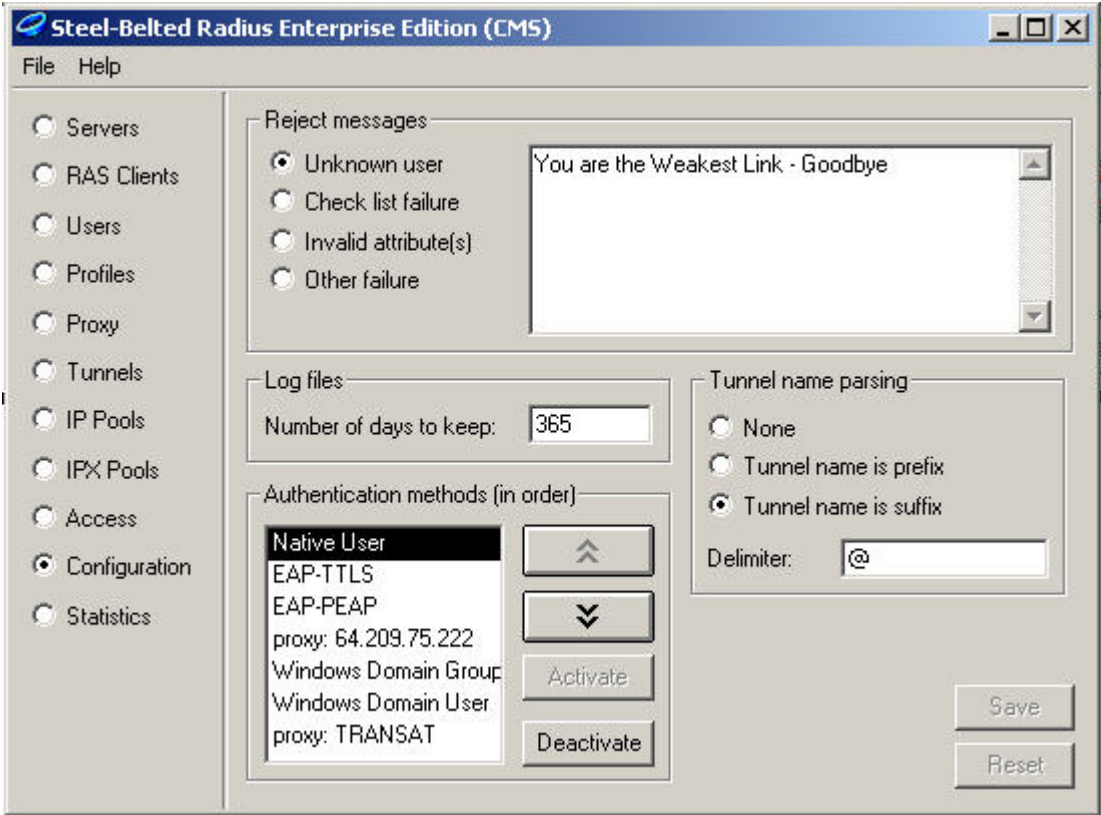


Figure B

Notice that “Tunnel name is suffix” is enabled for tunnel name parsing and that the delimiter is “@”. The delimiter must be “@” as that is only acceptable value on the USG II as a suffix delimiter.

Figure C is an example of a Steel-belted Radius tunnel profile showing the tunnel parameters used by the USG II to establish an L2TP tunnel.

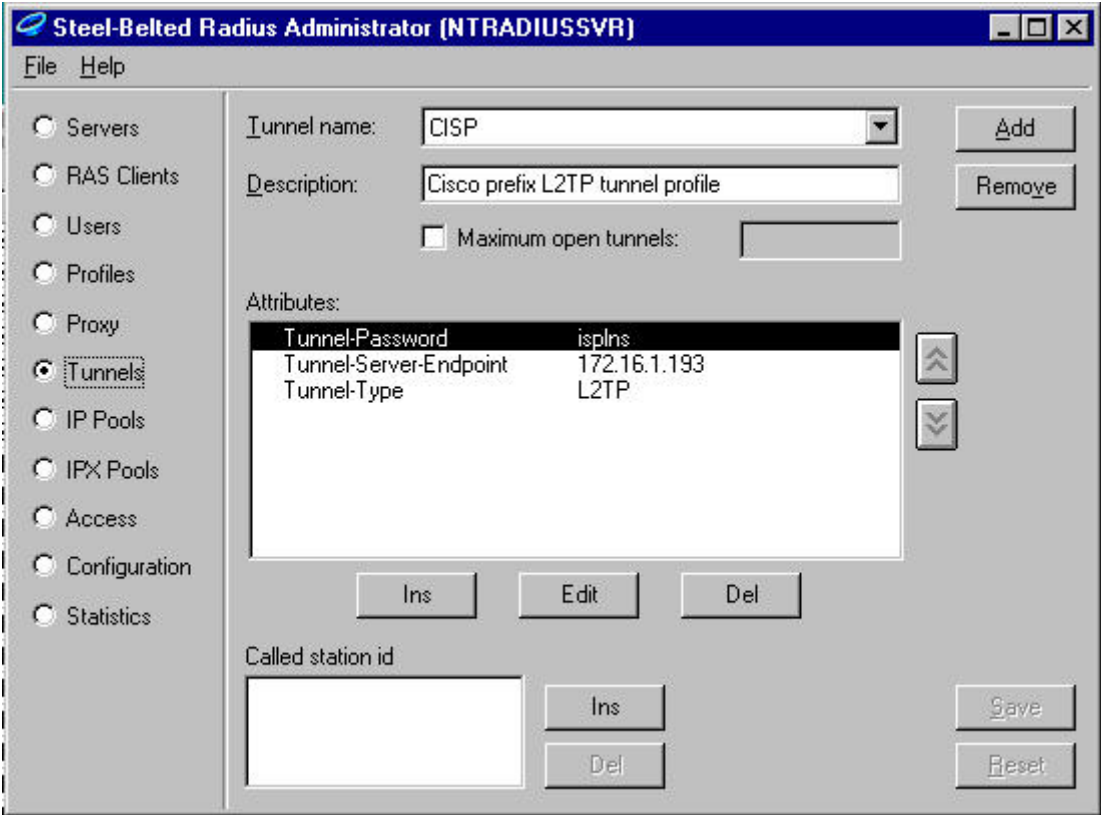


Figure C

This tunnel profile happens to handle prefix-based realms. In addition to the required Tunnel-Password and Tunnel-Server-Endpoint Radius attributes, it also contains the optional Tunnel-Type attribute. If the Tunnel-Type attribute is not supplied by the tunnel profile then the USG II will supply it during tunnel negotiation as that is the default and only acceptable value.

Suffix-based tunnel profiles are very much similar to prefix-based tunnel profiles. It is the Steel-belted Radius server's tunnel name parsing setting that determines whether the tunnel profiles are interpreted as prefix or suffix-based and matched to the realm-based usernames accordingly. However, the naming of prefix-based realms tends to be of the form "something/username" while suffix-based realms tend to be of the form "username@something.something".

Local Syslog and Syslog Filters

These settings can be accessed under the **Configuration/Logging** menu.

Log Settings

System Log	<input checked="" type="checkbox"/> Enable
System Log Number	<input type="text" value="7"/>
System Log Filter	<input type="text" value="7: Debug"/> ▼
System Log Server IP	<input type="text" value="67.130.148.58"/>
System Log save to file	<input checked="" type="checkbox"/> Enable

AAA Log	<input checked="" type="checkbox"/> Enable
AAA Log Number	<input type="text" value="7"/>
AAA Log Filter	<input type="text" value="7: Debug"/> ▼
AAA Log Server IP	<input type="text" value="67.130.148.58"/>
AAA Log save to file	<input checked="" type="checkbox"/> Enable

RADIUS History Log	<input type="checkbox"/> Enable
RADIUS History Log Number	<input type="text" value="0"/>
RADIUS History Log Filter	<input type="text" value="5: Notice"/> ▼
RADIUS History Log Server IP	<input type="text"/>
RADIUS History save to file	<input type="checkbox"/>

- 0: Emergency
- 1: Alert
- 2: Critical
- 3: Error
- 4: Warning
- 5: Notice
- 6: Info
- 7: Debug

Log Filter Setting:

The syslogs can be filtered at 7 levels as shown above. Setting the level to a number disables any syslogs above that filter setting. For e.g. setting the filter to 2:Critical only generates 0:Emergency, 1:Alert and 2:Critical level syslogs. All other syslogs are not generated.

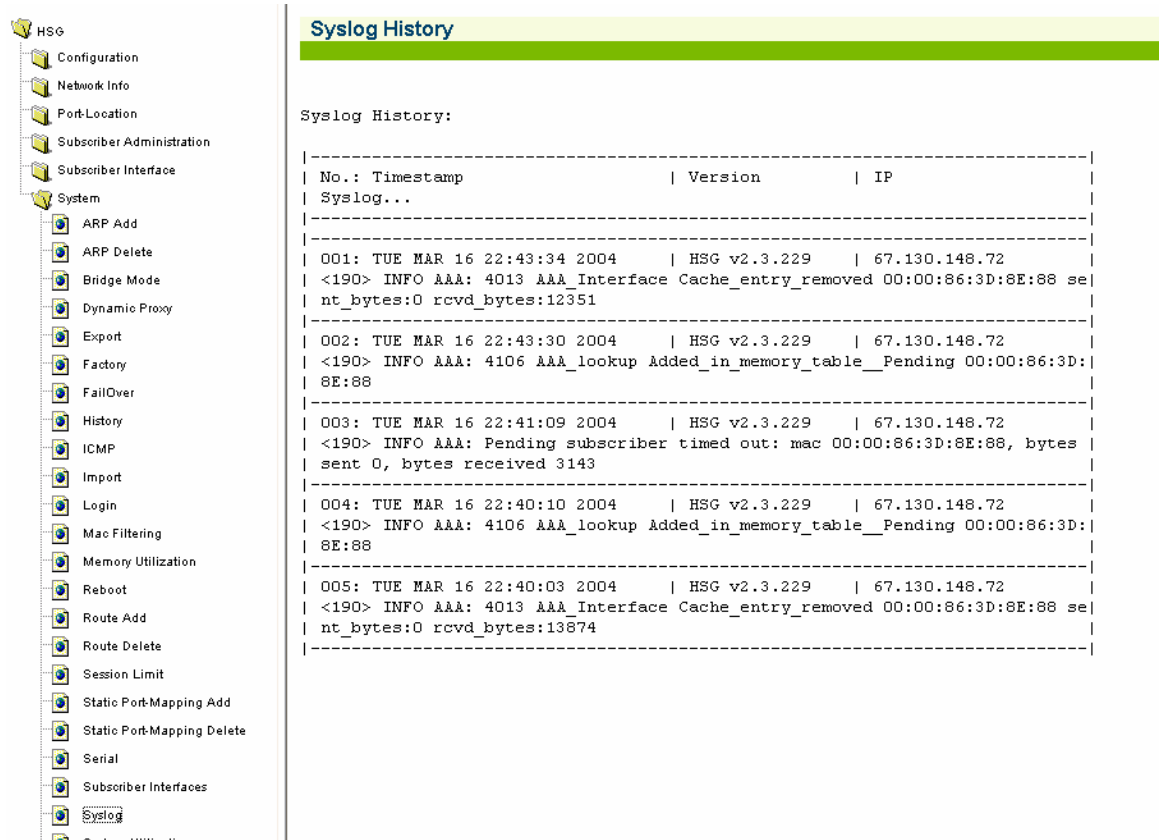
Log save to file Setting:

This setting enables/disables saving of syslogs generated by the system to a file named “syslog.txt” in the /flash directory of the NSE. This setting abides by the other settings set for the syslogs like filters, number and enable/disable.

It is not required to input a server IP address if you intend to only store the syslogs locally. Please leave the IP address field blank for such cases.

Warning: Do not configure the Server IP as the Network side IP of the gateway

Stored syslogs are viewable under **System/Syslog** menu. A total of 500 syslogs are stored locally.



Syslog History

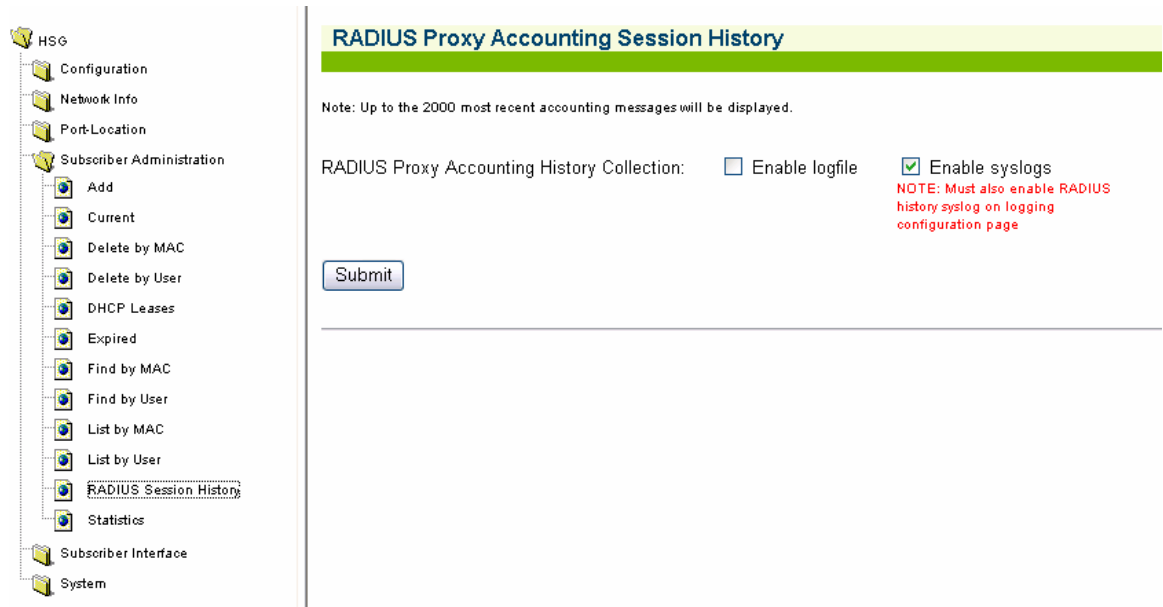
Syslog History:

No.:	Timestamp	Version	IP
001:	TUE MAR 16 22:43:34 2004	HSG v2.3.229	67.130.148.72
<190> INFO AAA: 4013 AAA_Interface Cache_entry_removed 00:00:86:3D:8E:88 sent_bytes:0 rcvd_bytes:12351			
002:	TUE MAR 16 22:43:30 2004	HSG v2.3.229	67.130.148.72
<190> INFO AAA: 4106 AAA_lookup Added_in_memory_table_Pending 00:00:86:3D:8E:88			
003:	TUE MAR 16 22:41:09 2004	HSG v2.3.229	67.130.148.72
<190> INFO AAA: Pending subscriber timed out: mac 00:00:86:3D:8E:88, bytes sent 0, bytes received 3143			
004:	TUE MAR 16 22:40:10 2004	HSG v2.3.229	67.130.148.72
<190> INFO AAA: 4106 AAA_lookup Added_in_memory_table_Pending 00:00:86:3D:8E:88			
005:	TUE MAR 16 22:40:03 2004	HSG v2.3.229	67.130.148.72
<190> INFO AAA: 4013 AAA_Interface Cache_entry_removed 00:00:86:3D:8E:88 sent_bytes:0 rcvd_bytes:13874			

Note: PageFaults are stored in the file named “lograw.txt” in the /flash directory and is not viewable on the web management interface.

RADIUS Proxy Accounting Logs

These settings are available under **Subscriber Administration/RADIUS Session History** menu.



"Enable Logfile" checkbox:

When this setting is enabled any RADIUS proxy accounting messages sent or received by the RADIUS proxy application are logged into a file named "RADHIST.RAD" in the /flash directory. This log contains accounting messages exchanged with downstream servers, and upstream NASs. The size of the log file is limited to 2000 records (accounting messages) or 320000 bytes -- when and if necessary the oldest records are purged to make room for new records.

If the logfile is disabled the current logfile is purged from the flash. If this is re-enabled again, only RADIUS accounting message sent/received from that point in time forward will be stored in the log.

"Enable Syslogs" checkbox:

If enabled then the same information described above is sent to the configured Syslog server. The content of the syslogs is sent in human-readable format.

The configuration page of the syslog server to which these RADIUS proxy accounting messages are sent is available under the Configuration/Logging menu as described above. The third and final set of Syslog parameters on that page pertains to the RADIUS History Log.

IPSec Tunneling

These settings are available under **Configuration/IPSEC** menu.

IPSEC Tunnel Settings

Global Settings

Enable IPSEC

Perfect Forward Secrecy

[Submit](#) [Reset](#)

Configured IPSEC Security Policies (up to 10 may be created)

SP#	Peer IP	Remote IP/Subnet	Remote Subnet Mask	Local IP/Subnet	Local Subnet Mask
There are 0 Security Policies at this time					

[Add](#) Click here to add a new IPSEC Security Policy.

NOTE: You must reboot for configuration changes to take effect.

[Reboot](#)

Enable IPSEC checkbox

When selected, enables the IPSEC tunnel mode functionality

Perfect Forward Secrecy checkbox

When selected, it enables PFS. PFS makes the keying material used in protecting the data independent of the keying material used for protecting the IKE exchanges.

New IPsec Security Policy can be added by clicking the Add button. It opens the following configuration page.

IPSEC Tunnel Security Policy Settings

Shared Key	<input type="text"/>
<hr/>	
Remote End	
Peer IP	<input type="text"/>
Remote IP/Subnet	<input type="text"/>
Subnet Mask	<input type="text"/>
<hr/>	
Local End	
<input checked="" type="radio"/> Use most current Network IP Address	
<i>Note: Network IP Address is dynamic if DHCP Client is enabled</i>	
<input type="radio"/> Custom Settings	
Local IP/Subnet	<input type="text"/>
Subnet Mask	<input type="text"/>

[Back to Main IPSEC Tunneling Settings page](#)

Shared Key setting

This is a secret shared between IPSec peers.

Remote End/ Peer IP setting

This is IP address of the remote VPN server.

The following settings define selectors of the Security Policy. All selectors must match in order for the policy to be applied.

Remote End/ Remote IP/Subnet setting

This is IP address of the remote network secured by the IPSec tunnel. The address could specify a host.

Remote End/ Subnet Mask setting

This is a subnet mask of the remote network secured by the IPSec tunnel

Local End/ Use most current Network IP Address radio button

Security Policy can derive the settings for the Local End from the current Network IP settings of the unit.

Local End/ Custom Settings radio button

Allows specifying custom settings for the Local End

Local IP/Subnet setting

This is IP address of the local network secured by the IPsec tunnel. The address could specify a host.

Subnet Mask setting

This is a subnet mask of the local network secured by the IPsec tunnel. The address could specify a host.

Remote Billing Mirror

The structure of the remote billing mirror record has been changed as follows:

- 1) Content-length has been added to the HTTP post.
- 2) The following two errors in the User Guide have been rectified
 - i. Length of REC_NUM was erroneously stated as 5 characters instead of 4 characters.
 - ii. SIGNATURE field was erroneously stated as a part of this record.

In the User Guide, under the Section “Mirroring Billing Records” and sub-section “XML Interface”

XML Interface

XML for the External Server

The USG II sends a string of XML commands according to specifications. HTTP headers are added to the XML packets that are built, as the billing “mirroring” information is sent to the external server in HTTP compliant XML format. The XML string built from the billing mirror record is in the following format:

USG II to External Server:

```
<USG RMTLOG_COMMAND="ADD_REC">  
<REC_NUM> max 4 characters </REC_NUM>  
<USG_ID> max 6 characters </USG_ID>  
<PROPERTY_ID> max 64 characters </PROPERTY_ID>  
<DATE> max 10 characters </DATE>  
<TIME> max 8 characters </TIME>  
<ROOM_NUM> max 20 characters </ROOM_NUM>  
<AMOUNT> max 10 characters </AMOUNT>  
<TRANS_TYPE> max 5 characters </TRANS_TYPE>  
<SIGNATURE> max 16 characters </SIGNATURE>  
</USG>
```

The new record with the content-length is as follows:

```
POST http://testing.com/brm HTTP/1.0
Content-Type: text/xml
Content-Length: 249
Host: 172.168.0.4

<USG COMMAND="ADD_REC">
  <REC_NUM>0000</REC_NUM>
  <USG_ID>012345</USG_ID>
  <PROPERTY_ID>USGII</PROPERTY_ID>
>
  <DATE>03/19/2004</DATE>
  <TIME>10:12:56</TIME>
  <ROOM_NUM>5</ROOM_NUM>
  <AMOUNT>1800.00</AMOUNT>
  <TRANS_TYPE>2</TRANS_TYPE>
</USG>
```