TOTAL CONTROL™

Security and Accounting

for SUN Solaris

INSTALLATION AND OPERATIONS GUIDE

Table of Contents

Chapter 1
Introduction

Accounting Server Overview

This feature is covered in detail in Chapters 3 and 4. The Solaris / UNIX version of the Accounting Server is distributed as part of the *TCM/Solaris* package, or packaged along with the Security Server.

The U.S. Robotics Accounting and Event Logging Server communicates with the Total Control chassis, NETServer Card, or other RADIUS-compliant devices to capture and log a wide variety of call accounting and event logging information to simple ASCII files. The Accounting Server's log files can be imported into most database applications.

You can configure which information the server should capture and maintain in the log files. The Accounting Server also supports over forty U.S. Robotics extensions to the RADIUS standard, for example, Call-End-Date-Time and Number-of-Rings-Limit. The Accounting Server logs incoming accounting and logging requests to three distinct log files:

- Total Control Call Accounting for NMC or NETServer
- Total Control Enterprise Network Events
- Native RADIUS Calls

Log Generation

The Accounting/Event Logging server sorts input requests and writes them to three separate tab-delimited logs. RADIUS event information is written to the Native RADIUS Call Log. NMC event records are separated according to event type: call-related event records are written to the Call Termination Log and all other NMC/Chassis event records are written to the Event Log.

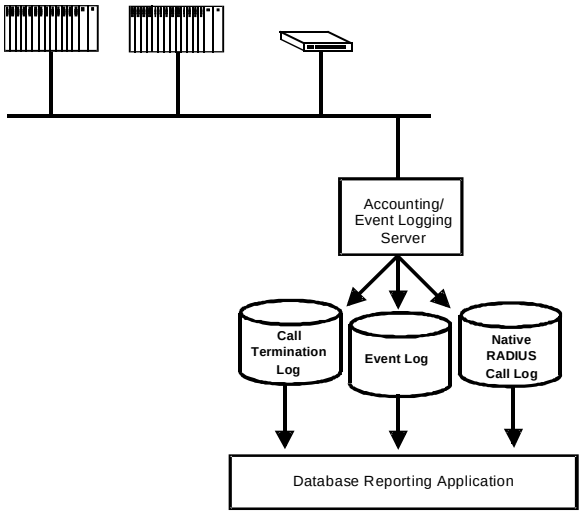

Figure 1-1. Log Files Generated

The log files generated by the server can be imported into any database application to be used for specific reporting purposes.

Security Server Overview

This feature is covered in detail in Chapters 5 and 6, and is distributed as an add-on option bundled with the Accounting Server.

The U.S. Robotics Security Server uses RADIUS authentication servers, clients and protocols to provide user security. Users connect to a Total Control chassis that either has a Network Management Card (NMC) with Security enabled, or a NETServer Card. The NMC or NETServer Card has an embedded RADIUS client that permits communication with a RADIUS authentication server.

Key features include, but are not limited to, the following:

- Support for the NETServer Card.
- Support for password aging and encryption.
- Support for case sensitive and minimum length passwords.
- Support for user blacklisting and user groups.
- Number of users limited only by disk space.
- NMC-only features include dialback groups, dial-in and dial-out call restriction lists.

How the Accounting Server and Security Server Interact

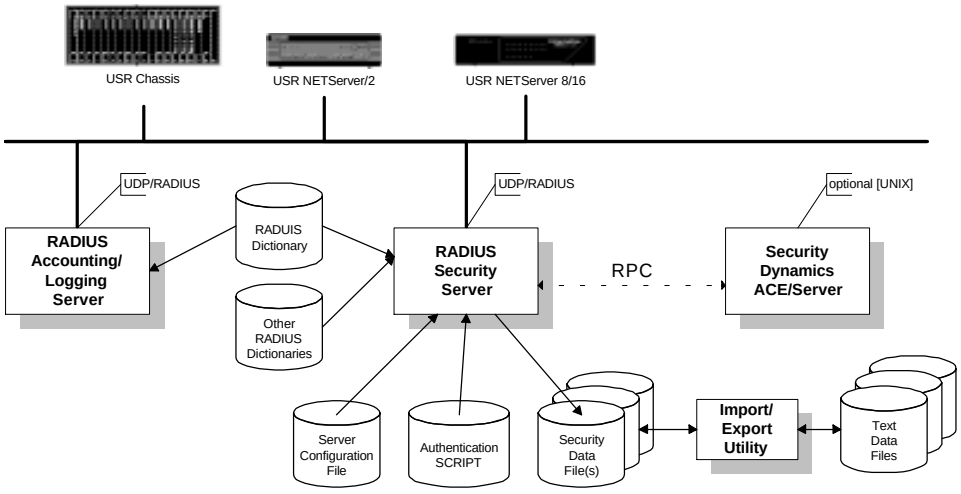

Figure 1-2. Server Interaction

RADIUS Client-Server Interaction

The RADIUS server communicates with the NMC and NETServer Card clients by polling UDP port 1645 for any client request.

- If a client is requesting a *password change*, the server identifies the client by searching the CLIENTS file for the Client ID. The server looks up the user name in the USERS file, and decrypts the user password using the MD5 algorithm (based on RSA Message Digest Algorithm).

  ⇒ If the decrypted user password matches the user password found in the USERS file, the new password is updated and the server sends the password accept back to the client.

  ⇒ A password reject is returned to the client if the Client ID is not found in the CLIENTS file, or the user name is not found or is different than that in the USERS file.

- If the client is sending an authentication request, the server identifies the client and user and decrypts the user password.

  ⇒ If Client ID, user name or user password cannot be verified, the server sends an authentication reject to the client.

  ⇒ If the Client ID, user name and user password are verified, the server double-checks all entries in the request, and if they are valid, the server responds to the client with an authentication accept.

> **NOTE:** If a user is set up for both the NMC and the NETServer clients, an authentication accept response from the server to the client contains attributes for both devices. Attributes that are unknown to a client are ignored.

Whether the response is *accept* or *reject*, a character string may also be sent back to the server for any of the following reasons:

- The user is within the Password Warning period.
- The user name could not be found or the MD5-calculated password does not match the server password for a user.
- The user is currently blacklisted.
- The user's password has expired.
- The Client ID sent by the client was unknown to the server.

RADIUS Database

The RADIUS database exists on the server as a set of files you can edit. The database itself exists in ASCII format where records contain multiple fields with which the server controls the authentication process and additional security exchanges. RADIUS database files are described in Appendix D.

Chapter 2
Installation

NOTE: The Accounting Server and Security depend upon the following shared libraries normally present in Solaris 2.4:

/usr/lib/ld.so.1
/usr/lib/libsocket.so.1
/usr/lib/libnsl.so.1
/usr/lib/libdl.so.1
/usr/lib/libm.so.1
/usr/lib/libC.so.1
/usr/lib/libw.so.1
/usr/lib/libc.so.1
/usr/lib/libintl.so.1

Installing the Accounting Server

The Accounting Server is distributed as part of the *TCM/Solaris* package or packaged along with the Security Server.

If you have purchased the Accounting Server as part of *TCM/Solaris*, you do not need to follow the steps listed below. The *TCM/Solaris* installation script updates the run states to start the Accounting Server automatically whenever you start the workstation it is installed on. It also updates the run states to stop the Accounting Server automatically when the workstation is shut down.

If you have purchased the Accounting Server along with the Security Server, you must perform the steps listed below.

1. Log in as **root**.
2. Specify where to install the Accounting Server. If you have already installed other *TCM/Solaris* utilities, use the same directory you specified for them. The default directory is /tcm.

- *C shell users:* Type the following command (also consider adding it to your **.cshrc** file):

  # setenv TCMHOME /tcm

- *Bourne shell users:* Type the following commands (also consider adding them to the **.profile** file):

  # TCMHOME=/tcm
  # export TCMHOME

3. From a UNIX shell/command line, create the TCM home directory ($TCMHOME) using the **mkdir** command. Then change to the new directory using the **cd** command.

   # mkdir -p $TCMHOME
   # cd $TCMHOME

4. Insert the floppy, mount it, and tar the files onto the disk.

   # volcheck
   # tar -xvf/floppy/accting/accting.tar

5. Execute the install script.

   # cd $TCMHOME/INSTALL
   #./accting

   This script updates the run states: when the UNIX box is booted, the Accounting Server starts automatically. When the UNIX box is shut down, the Accounting Server also shuts down automatically.

NOTE: If the installation machine is an NIS client, the following lines may have to be added manually to the NIS server's /etc/services file:

radacct	**1646/udp**	**# radius accounting server**

Accounting Server Commands

- To run the Accounting Server, do the following:

  # cd $TCMHOME/bin
  # ./accting start

- To stop the Accounting Server, do the following:

  # cd $TCMHOME/bin
  # ./accting stop

- To restart the Accounting Server, do the following:

  # cd $TCMHOME/bin

```
# ./accting restart
```

Brief Configuration Information

The two main configuration files for the Accounting Server are:

```
$TCMHOME/config/accting/outdef.dat
$TCMHOME/config/common/dictnary.dat
```

These files contain enough default values so that the Accounting Server can run and can store accounting information into files stored in $TCMHOME/log. See Chapter 4 for details on customizing the files.

Uninstalling the Accounting Server

You can choose to uninstall the Accounting Server completely, or leave the configuration and log files intact.

Uninstalling the Accounting Server and All Files

1. Log in as **root**, and remove the directories and files.
   - *C shell users:* Type the following commands:

     ```
     setenv TCMHOME/tcm
     # cd $TCMHOME
     # ./Remove -c
     ```

   - *Bourne shell users:* Type the following commands:

     ```
     # cd $TCMHOME/REMOVE
     #./accting -p
     ```

2. Remove the path statements from the C and/or Bourne shells.

Uninstalling the Accounting Server Only

1. Log in as **root**, and remove the directories and files.
   - *C shell users:* Type the following commands:

     ```
     setenv TCMHOME/tcm
     # cd $TCMHOME
     # ./Remove -c
     ```

   - *Bourne shell users:* Type the following commands:

     ```
     # cd $TCMHOME/REMOVE
     #./accting -p
     ```

2. Remove the path statements from the C and/or Bourne shells.

Installing the Security Server

1. Log in as **root**.
2. Specify where to install the Security Server. If you have already installed other *TCM/Solaris* utilities, use the same directory you specified for them. The default directory is /tcm.
   - *C shell users:* Type the following command (and consider adding it to your **.cshrc** file):

     # setenv TCMHOME /tcm
   - *Bourne shell users:* Type the following commands (and consider adding them to the **.profile** file):

     # TCMHOME=/tcm
     # export TCMHOME
3. From a UNIX shell/command line, create the TCM home directory ($TCMHOME) using the **mkdir** command. Then change to that directory using the **cd** command.

   # mkdir -p $TCMHOME
   # cd $TCMHOME
4. Insert the floppy, mount it, and tar the files onto the disk.

   # volcheck
   # tar -xvf/floppy/security/security.tar
5. Execute the install script.

   # cd $TCMHOME/INSTALL
   #./security

   This script updates the run states so that when the UNIX box is booted, the Security Server starts automatically. When the UNIX box is shut down, the Security Server also shuts down automatically.

NOTE: If the installation machine is an NIS client, the following lines may have to be added manually to the NIS server's /etc/services file:

radserv	**1645/udp**	**#radius security server**
securid	**755/udp**	**#ACE security server**
securidprop	**5510/tcp**	**#ACE security salve server**

Security Server Commands

- To run the Security Server, do the following:

  # cd $TCMHOME/bin
  # ./security start

- To stop the Security Server, do the following:

  # cd $TCMHOME/bin
  # ./security stop

- To restart the Security Server, do the following:

  # cd $TCMHOME/bin
  # ./security restart

Brief Configuration Information

The main configuration files for the Security Server are:

$TCMHOME/config/security/radserv.cfg
$TCMHOME/config/security/radserv.scp
$TCMHOME/config/security/usrsec.dic
$TCMHOME/config/common/dictnary.dat

After installation, these files contain enough default values so that the Security Server can run. However, you may want to edit RADSERV.CFG and RADSERV.SCP to fit your needs. See Chapter 6 for details on customizing these files. DICTNARY.DAT and USRSEC.DIC should not be modified.

Additionally, there are four security data files on users, RADIUS clients, user groups, and dialback groups. See Chapter 6 for details on adding users, user groups, and so on to these files.

$TCMHOME/config/security/users.txt
$TCMHOME/config/security/clients.txt
$TCMHOME/config/security/groups.txt
$TCMHOME/config/security/dbgroups.txt

Uninstalling the Security Server

You can choose to completely uninstall the Security Server, or leave the configuration and log files intact.

Uninstalling the Security Server and All Files

1. Log in as **root** and remove the directories and files.
   - *C shell users*: Type the following commands:

     setenv TCMHOME/tcm
     # cd $TCMHOME
     # ./Remove -c

   - *Bourne shell users:* Type the following commands:

     # cd $TCMHOME/REMOVE
     #./security -p

2. Remove the path statements from the C and / or Bourne shells.

Uninstalling the Security Server Only

1. Log in as **root** and remove the directories and files.
   - *C shell users:* Type the following commands:

     setenv TCMHOME/tcm
     # cd $TCMHOME
     # ./Remove -c

   - *Bourne shell users:* Type the following commands:

     # cd $TCMHOME/REMOVE
     #./security -p

2. Remove the path statements from the C and / or Bourne shells.

Chapter 3
Accounting Server

Traps vs. Logs

By setting up SNMP traps in Total Control, you ensure that certain system events will trigger a system response of some kind. Traps can be used by SNMP management software in a variety of ways, such as to trigger alarms or to launch programs. Another way to use this information is to trigger a log entry by having the NMCand / or NETServer client forward it to the Accounting Server.

The Accounting Server's clients communicate with the server via Ethernet or Token Ring IP LAN. Information such as call statistics and specific chassis events may be sent by clients across a LAN connection to be logged. You may then use a post-processing application or database script to format the logged data.

The *enableAll* trap MIB enumeration setting will send out data both as a trap and as a log entry (see the note below). Depending on how you set up the server, it will format some or all of the data received from the client(s) into logs you can view and analyze at a later time.

NOTES:

- We recommend that users do not send log data to a server via a SLIP connection.
- We recommend the Accounting Server be on a dedicated workstation and discourage locating it on the same workstation to which traps are being sent.

- You must use a text editor to configure the Accounting Server.

Files and Logs

Input Files

The Accounting Server uses two ASCII files to define how it functions: a Dictionary file and an Operational Definition (or Database Definition) file.

Dictionary File (DICTNARY.DAT)

The dictionary file matches attribute numbers to their type and a textual representation. The Dictionary file contains all RADIUS and USR extensions needed for both the Accounting and Security Server. The file is located in the common directory used by both the Accounting and Security applications.

Below is an example of a standard RADIUS attribute:

```
ATTRIBUTE    User-Service-Type    6    integer
```

The attribute can be further described using a VALUE entry later in the file. For example:

```
VALUE    User-Service-Type Login-User     1
VALUE    User-Service-Type Framed-User    2
```

The USR implementation extends the dictionary file to include USR-specific attributes and provide output formatting capability. Native RADIUS attributes and USR-specific attributes belong to the same name-space (that is, there is not a USR-specific attribute *User-Name*). The server application treats all attributes equally after they are extracted from the request packet. The Attrib-Number for USR attributes is a 16-bit value rather than the RADIUS 8-bit value. The datatype is limited to the original RADIUS specification (i.e., integer, string, IP address, date). An additional keyword, ATTRIB_NMC, is used to define USR-specific attributes. The Attribute Name is given, the Attribute Number, Datatype, and Format. For example:

```
ATTRIB_NMC    Chassis-Slot    0xBF39    integer    Slot-%
```

In the above example, if a value of 10 is returned in the USR Slot-Index attribute, it is written to the database/log file as “Slot-10”.

The DICTNARY.DAT file is located in the /accting subdirectory of the TCM home directory you specified when installing the Accounting Server. The DICTNARY.DAT file should not be edited, but may be referred to for the names of attributes to be programmed into the OUTDEF.DAT file.

Output Definition File (OUTDEF.DAT)

This file is an external database definition file that defines the format of each of the log files. The file maps attributes (defined in the dictionary file) to specific column (field) positions in the output file. In addition, this file specifies:

- The method of time stamping the log file entries (TIME)
- The directory where log files should be written to (PATH)
- What action(s) should be taken when the disk is low on space or full (DISK_LOW and DISK_FULL)
- What action(s) should be taken when an old log file is closed and a new log file is opened (NEW_LOG_FILE).

The OUTDEF.DAT file is located in the /accting subdirectory of the TCM home directory you specified when installing the Accounting Server. See Chapter 4 for information on configuring the OUTDEF.DAT file.

Output Files

The Accounting/Event Logging application operates on one set of files for a 24-hour period. At midnight, the server closes and re-initializes the files. This means that log files are not available for import on the same day they are generated. The files are named using the format *yymmdd.ext*, where yy = year, mm = month, dd = date, and ext is the three character extension defined for the file type (typically .con, .nmc, or .rad).

If the server runs out of available disk space, the application is closed and the clients are forced to switch to an alternate server. To prevent this from happening, an administrator should set up scripts for NEW_LOG_FILE and DISK_LOW to delete log files when they are no longer required.

The Accounting server sorts input requests and writes them to three separate output files. Those files are the Call Termination Log (.con), Event Log (.nmc), and Native RADIUS Call Log (.rad). Generic RADIUS and NMC generated requests are separated by the presence of a USR vendor-specific attribute Event-Id. NMC requests are then further separated into event log records and call termination records.

NOTE: Every request has an additional attribute added by the server itself, Server-Time (the time that the request was received).

Call Termination Log (.con)

Depending on your configuration of the NMC / NETServer client, the Call Termination Log collects call termination records generated by the chassis modems. The data is formatted in the log according to your customization of the OUTDEF.DAT file in the NMC_CONNECTIVITY section. The table format may have a column assigned for each possible attribute (both RADIUS and USR) that can be sent for these events.

An NMC can be configured to send additional information, depending on user requirements. This additional information is based on four different groups of statistics. The current default configuration sends messages from the first group, Usage Statistics. Usage Statistics and the other groups are listed in Appendix C under *Call Termination Log*.

These statistics are combined with the Generic Slot record and the Entity Index. The default log fields are listed below:

Common Name	Attribute Name	Format	Example
IP	Client-Id	IPdot	111.222.133.244
Slot	Chassis-Slot	%	11
Channel	Channel	%	2
TimeStamp	Event-Date-Time	Date1	02/01/95 13:49:09
EventID	Event-Id	%	1
Sequence	Acct-Session-Id	%	1004
Call Start Time	Call-Start-Date-Time	Date1	02/01/95 13:15:20
Call End Time	Call-End-Date-time	Date1	02/01/95 13:46:12
DiscCode	Connect-Term-Reason	%	4
FailCode	Failure-to-Connect-Reason	%	19
bpsCode	Default-DTE-Data-Rate	%	8

You can also configure the Call Termination Log to include the following fields:

- Acct-Session-Id
- Acct-Session-Time
- Acct-Status-Type
- Client-Port-Id

- Dialback-No
- Last-Caller's-Number-ANI
- Last-Number-Dialed-In-DNIS
- Server-Time
- User-Name

The mechanism for collecting and formatting these records is to place each attribute from the input packet into a column defined by output definition file. If an attribute is located that does not have an associated column, it is silently discarded. Any "empty" columns are filled with the delimiting or tab character.

Event Log (.nmc)

Depending on your configuration of the NMC client, the Event Log collects all NMC event records sent to the server, except call termination records. This data is formatted in the log according to your customization of the OUTDEF.DAT file, and is covered in the section called NMC_EVENTLOG. The mechanism for collecting and formatting Event Log records is basically the same as for the Call Termination Log, except that if an attribute is sent that does not have an associated column, it is formatted using the dictionary and appended to the table's final column (not listed in the output definition file). It is, therefore, important to have dedicated columns for all regularly sent attributes, otherwise they are placed in this default column.

This method allows all chassis event records to exist in a single table. If set up correctly, the last column will represent a textual description of the event and relevant values. For example, a Module-Inserted event might give the card description V.34_Dual_Modem.

Common Name	Attribute Name	Format	Example
IP	Client-Id	IPdot	111.222.133.244
TimeStamp	Event-Date-Time	Date1	02/01/95 13:49:09
EventID	Event-Id	%	1
Sequence	Acct-Session-Id	%	1004
Message	{the formatted event message}		"Slot-% Channel-%"

The Message is a concatenation of all other USR-specific attributes in the RADIUS packet.

Native RADIUS Call Log (.rad

The Native RADIUS Call Log collects all native RADIUS standard attributes. Characteristics of this file are defined in the OUTDEF.DAT file as GENERIC_RADIUS

Common Name	Attribute Name	Format	Example
IP	Client-Id	IPdot	111.222.133.244
TimeStamp	{internal PC time of server}	Date1	02/01/95 13:49:09
Event-Id	Acct-Status-Type	%	1
Session-Id	Acct-Session-Id	%	1004
Connect-Time	Acct-Session-Time	%	36

Event-Id is set to either 1 (Start) or 2 (Stop).

Session-Id is the same for the Start and Stop RADIUS packets.

Connect-Time is given in seconds.

Chapter 4
Accounting Server Configuration

To configure the Accounting Server, you need to edit the Output Definition file, OUTDEF.DAT. The dictionary file, DICTNARY.DAT, does not require configuration.

The OUTDEF.DAT file is the mechanism that permits you to control the content of the three logs. By editing the OUTDEF.DAT file, you can exert a high level of control over the content of the logs as well as automate actions to take at different stages of log creation. You may use your preferred text editor to edit the OUTDEF.DAT file so it will generate and customize desired logs.

The Output Definition File

File Syntax

- In order to generate logs, you must edit the OUTDEF.DAT file using the correct syntax. There are three components to this syntax that need to be understood: keywords, attributes, and format specifiers. See Appendix C for complete information.
- If the first field of a line does not contain one of these keywords, it is parsed as an attribute. Attributes are contained in the dictionary file (DICTNARY.DAT). The attributes that follow a line containing the keyword FILE indicate that the values for those attributes are to be stored in a log file (.con, .rad, or .nmc).
- The format specifier is an optional element of the attribute line that allows you to override the default format that is used to store the attribute value in the log.

Configuring the OUTDEF.DAT File

See Appendix C for a complete description.

1. From an ASCII text editor, open OUTDEF.DAT.
2. Configure the Time Zone (TIME).

   Valid options are GMT (Greenwich Mean Time) or LOCAL. For example:

   TIME GMT

3. Define the directory where the log files will be written to (PATH).

   You must specify the keyword PATH and the full directory path. The directory must already exist. For example:

   PATH ./files

   NOTE: The example specifies a subdirectory (indicated by the "/") named "files" that is under the directory from which the Accounting Server is launched (indicated by the "."). If the server were to be launched from a different directory, the path indicated in the example may be invalid and the file may not be created.

4. Define what action(s) the Accounting Server should take when free disk space is at a minimum (DISK_LOW).

   You must create a script or other executable file that defines those actions, for example, delete all log files more than one month old. It must be in the same directory as the log files.

   You must also specify the minimum free disk space threshold in Mbytes. If the DISK_LOW keyword does not appear, a default of 10 Mb is assigned for the free disk space threshold. For example:

   DISK_LOW disklow.bat 12

   Disk space is checked periodically, and when the free disk space threshold reaches 12 Mb, DISKLOW.BAT is executed.

5. Define what action(s) the Accounting Server should take when free disk space is at a fatal level (DISK_FULL). This event may occur if DISK_LOW fails to increase the free disk space.

   You must create a script or other executable file that defines those actions, for example, delete all log files more than one month old. It must be in the same directory as the log files. After the file is executed, the Accounting Server terminates.

You must also specify the fatal free disk space threshold in Mbytes. If the DISK_FULL keyword does not appear, a default of 5Mb is assigned for the free disk space threshold.

Typically, the executable is used to initiate an alarm of some kind, page an administrator, etc. If DISK_FULL does not appear in the file, a default of 5 Mbytes is assigned for the fatal free disk space limit.

For example:

DISK_FULL diskfull 6

When free disk space falls to 6 Mb, DISKFULL.BAT is executed.

6. Define what action(s) the Accounting Server takes when the log files are closed at midnight each night (NEW_LOG_FILE). Whatever you specify is executed for each log file that is closed at midnight.

   You must create a script or other executable file that defines those actions. It must be in the same directory as the log files.

   At midnight, each log file is closed and new log files reflecting the new date are created. After the new log files are created, the specified file is executed using the full path name of the log files as a parameter. This file typically purges the old log files and/or imports the old log file into a database. For example

   NEW_LOG_FILE newlog.bat

7. Define the three Accounting Server log files (FILE).

   Each log file must have the keyword FILE followed by the name of each log file, for example, NMC_CONNECTIVITY for the Call Termination Log. Note that each log file will actually be the date in the format yymmdd followed by "." and the suffix (con, nmc, or rad).

   On the same line, you must cite the log files' suffix (three character limit) that indicates type of information stored (.con, .rad, .nmc).

   You may then define what attributes are logged to this file and what output format they will have, each on a separate line. For example:

```
FILE    NMC_CONNECTIVITY        con
        Client-Id               IPdot
        Chassis-Slot            %
        Channel                 %
        Event-Date-Time         Date1
        Event-Id                %
        Call-Start-Date-Time    Date1
        Call-End-Date-Time      Date1
        Connect-Term-Reason     %
        Default-DTE-Data-Rate   %
```

Sample OUTDEF.DAT File

```
#       OUTDEF.DAT  --  Sample Output Definition file

TIME                    LOCAL
PATH                    ./tcm/bin/accting/files
DISK_LOW                disklow.bat         12
DISK_FULL               diskfull.bat        6
NEW_LOG_FILE            newlog.bat

FILE        NMC_CONNECTIVITY            .con
            Client-Id                   IPdot
            Chassis-Slot                %
            Channel                     %
            Event-Date-Time             Date1
            Event-Id                    %
            Call-Start-Date-Time        Date1
            Call-End-Date-Time          Date1
            Connect-Term-Reason         %
            Default-DTE-Data-Rate       %

FILE        NMC_EVENTLOG                .nmc
            Client-Id                   IPdot
            Chassis-Slot                %
            Channel                     %
            Event-Date-Time             Date1
            Event-Id                    %
            Acct-Session-Id
            Failure-to-Connect-Reason   %
            Server-Time                 Date1
            Client-Port-Id              %

FILE        GENERIC_RADIUS              .rad
            Client-Id                   IPdot
            Server-Time                 Date1
            Acct-Status-Type            %
            Acct-Session-Id
            Acct-Session-Time           %
```

Chapter 5
Security Server

General Overview

The Security feature uses the client-server protocol RADIUS (Remote Authentication Dial-In User Service) to provide user authentication. RADIUS is a public domain protocol that regulates access to a secure network through a centrally managed server. You may do the following:

- Use the authentication server for USR modems and NETServer
- Use the authentication server for non-USR products
- Use a single authentication server for multiple Total Control chassis, as well as multiple non-USR products
- Use previously created RADIUS database files

No hardware changes are needed, but Security does require compatible versions of the NMC, *Total Control Manager*, and modem software.

The graphic below demonstrates how the different components of this feature operate together.

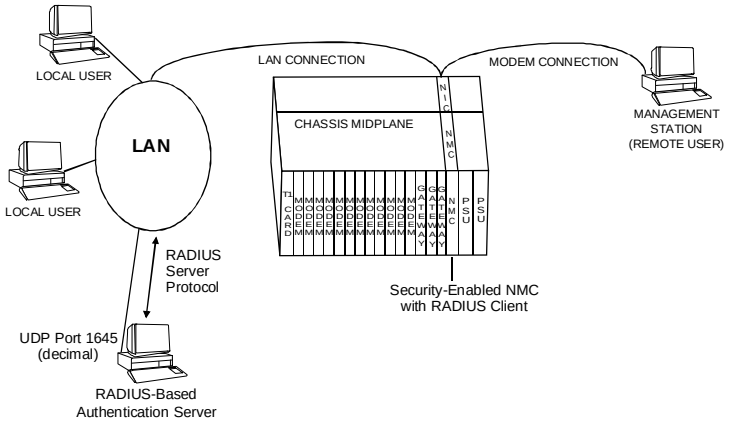

Figure 5-1. Security Operation

Security allows you to prevent unauthorized users from making outbound (Dial Out) calls using rack modems, and making inbound (Dial In) connections with these modems. Both Dial In and Dial Out calls generate events to which the system responds according to preconfigured actions.

Security implements a User Database to store security configurations on a per-user or group basis, and uses the Accounting Server to record security-related events.

RADIUS Compatibility

RADIUS is a protocol defined in the Network Working Group Internet draft (DRAFT-IETF-NASREQ-RADIUS-03.TXT). Based on a model of distributed security previously defined by the Internet Engineering Task Force (IETF), RADIUS provides an open and scaleable client-server security system.

Internet Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or rendered obsolete by other documents at any time. It is inappropriate to use Internet Drafts as reference material or to cite them other than as "work in progress."

To learn the current status of any Internet Draft, please check the "1id-abstracts.txt" listing contained in the Internet Drafts Shadow Directories on ds.internic.net (U.S. East Coast), nic.nordu.net (Europe), ftp.isi.edu (U.S. West Coast), or munnari.oz.au (Pacific Rim).

NETServer Security Feature Overview

Using this feature, a network administrator can maintain a single user table for all NETServers on the network, rather than individual tables for all cards. Each NETServer acts as a client of the RADIUS server.

When a user dials in to the NETServer, the NETServer first checks its own Users Table. If it can't find the user, and it is configured to do so, it sends an authentication request to the RADIUS server.

The NETServer encrypts the password using an encryption key shared by the NETServer and the RADIUS server, and passes the user name and the encrypted password on to the RADIUS server. The RADIUS server then checks the user name and password against its USERS file, accepts or rejects the user, and passes this information back to the NETServer client.

If access is denied, the NETServer disconnects. If access is granted, the RADIUS server forwards the appropriate user table information to the NETServer. Configuration of this information.

Hub Security Feature Overview

In a typical configuration, customers connect to a Total Control chassis that has a security-enabled NMC. The NMC has a RADIUS client embedded in it, and communicates through a LAN to a RADIUS authentication server. This server is a USR-proprietary RADIUS server. During a Security session, the modem does the following.

- Allows the NMC to conduct a dialog with the user (the login sequence) prior to completing any connection. This requires routing Security data to the NMC.
- Reports security events.
- Initiates the standard DTE RS-232 handshaking sequence after the login sequence has been successfully completed.

Dial Out Security

Total Control provides two levels of Dial Out Security: user validation and call restriction. When a modem is configured to support Dial Out Security and the NMC is present, all Dial Out calls are intercepted by Total Control. The system requires users to enter their name and password before a call is placed. Upon successful completion of the login sequence, Total Control compares the phone number that the user has dialed against the Call Restriction List that is assigned to that user.

The Call Restriction List is a table that identifies numbers that may or may not be called by the user. For example, all 900 numbers may be excluded. If a number is allowed, Total Control permits the call to be placed.

A Dial Out Security call follows this sequence.

1. The modem reports to the NMC that a dial out call has been requested on any attempt to originate a call from the DTE. This includes both ATD commands and calls dialed automatically when the DTE asserts the Data Terminal Ready signal (DTR).
2. The modem stores the requested number as the last number dialed.

3. The NMC queries the modem for the requested number and conducts the user login sequence by prompting for name, password, etc. If name/password are invalid, the session is terminated.
4. The NMC evaluates the number against the Call Restriction List and then instructs the modem to either complete the phone call or to hang up and return to command mode, effectively terminating the dial out attempt.

Dial In Security

When a Total Control rack modem is configured to support Dial In Security and the NMC is present, all incoming calls are intercepted by the Total Control system. Dial-in users must enter a valid name and password before the connection to the DTE is permitted.

A Dial In Security call follows this sequence.

1. The modem detects an incoming ring and completes the phone line connection to the remote modem.
2. The modem reports to the NMC that a dial in attempt has been made.
3. The NMC and the modem perform the appropriate Security login sequence, depending on the mode of Dial In Security that has been configured for that particular user (see the next section).
4. When the Security login sequence is completed successfully, the NMC instructs the modem to ring the local DTE.
5. If the login sequence is unsuccessful, the NMC instructs the modem to terminate the connection.

Dial In Security Modes

You can configure a user for one of three different, mutually exclusive Dial In Security modes. These modes take effect after the dial-in user has successfully completed the initial login sequence.

Pass-Through Mode

The call is passed through to the DTE. The modem's DTE interface is handled just as if a normal call is being made. The modem sends the DTE the EIA control signal Ring Indicate and waits for the DTE to respond.

NOTE: The modem must be configured to S0=1 or higher. The modem will not respond to AT commands (for example, ATA) at this point in the setup.

Dial Back Stored Number

After the name and password are validated, the call is terminated and Total Control calls the user back at a previously assigned number. When the user is called back, you can optionally configure the system to prompt users to reenter their name and password, as an additional check. After this sequence is completed, the call is passed through to the DTE as described in Pass-Through mode.

Dial Back Entered Number

After the name and password are validated, users are prompted to enter a phone number at which to be called back. You can customize the prompt for this number. The call is then terminated and the user is called back at the entered number, at which point you can again require the user to enter name and password. Before dialing back the entered number, you can also configure the system to check it against a Call Restriction List. After this sequence is completed, the call is passed through to the DTE as described in Pass-Through mode.

Assigning a Dial Back Modem

If you configure Dial In Security to call back the user at a stored or entered number, the NMC must reserve a modem to dial back. This can be the same modem used for dial in or a different one. Optionally, present the user with a menu listing the allowed dial back modems.

Blacklisting

Both Dial In and Dial Out Security incorporate the concept of blacklisting. The purpose of blacklisting is to limit access to Total Control modems to users with valid user names and passwords. This is accomplished by setting a finite number of login attempts. Users who fail to enter a correct name/ password combination within a preconfigured number of consecutive login attempts are denied access to the system. After a user logs in successfully, the login attempts counter is reset to zero.

The number of permissible login attempts may be set from 0 to 15, with 0 meaning unlimited login attempts. Once users are blacklisted, they cannot access the system again until their user configuration is reset in the Security database.

Chapter 6
Security Server Configuration

Security Server Files

The Security Server has several files that you need to configure:

- The Security Configuration File (RADSERV.CFG). This file defines the operating characteristics of the Security Server, such as clients, dictionary files, and so on.
- The Security Data Files. These are clear text files that contain the information necessary to authenticate security requests, and include the Users, Groups, and Dialback Groups.
- The Security Authorization File. This is an ASCII script file that defines the authentication process.

You can use a simple ASCII text editor to configure or edit these files, or to import security information from other files.

Configuring the Security Configuration File

See Appendix D for a complete description of each parameter.

1. From an ASCII text editor, open RADSERV.CFG.
2. Configure the **[SYSTEM]** parameters.

   Define the main Security Authorization script using the SCRIPT variable. Remember to include the full path name. For example:

   Script=/tcm/security/radserv.scp

   Configure the password characteristics: how many days a password is valid, how many days before expiration users are warned, and the minimum number of characters required.

For example:

Password-Expiration=90
Password-Warning=7
Password-Minimum=5

Define the blacklist characteristics: whether the blacklisting is enabled (1) or disabled (0), and the number of failed logins permitted before a user is blacklisted. For example:

BlackList=1
BlackList-Threshold=3

If the Security Server will pass authentication requests to a Security Dynamics ACE/Server, define whether the NMC and NETServer clients support RADIUS Access-Challenge. ACE$Challenge-Support=0 is Off and ACE$Challenge-Support=1 is On.

Also, configure the amount of time in seconds before the Security Server declares the ACE/Server unreachable. For example:

ACE$Challenge-Support=1
ACE$Server-Timeout=45

Define what events you want logged to the USR Accounting Server's log file(s). 0=Event Not Logged and 1=Event Logged.

Log-Failed-Logins=1
Log-Successful-Logins=0
Log-User-Blacklisted=1
Log-User-Password-Change=1
Log-Security-Breach=1

3. Define the RADIUS dictionary files that the Security Server will use in the **[DICTIONARIES]** section. You can specify more than one dictionary, but at least one is required. For example:

   1 = /etc/raddb/usrsec.dic
   2 = /etc/raddb/dictnary.dat

4. Specify all security data files used by the Security Server in the **[TABLES]** section.

   Each security data file is given a table-name referenced in the authorization script. Do not forget to include the full path of the file(s). For example:

   USERS = users.txt
   CLIENTS = clients.txt
   DBGROUPS = dbgroups.txt
   GROUPS = groups.txt

5. Specify the other RADIUS and / or Accounting Servers in the **[SERVERS]** section.

   Each server is identified by a Server-Id. An IP address and UDP port is associated with that Server-Id. The USR Accounting Server must be given the Server-id of "Accounting". The IP address must be in dot notation, and the UDP port is usually 1646. For example:

   Accounting.ip = 192.77.123.156
   Accounting.port = 1646

6. Save the file.

Sample Configuration File

```
#       RADSERV.CFG          --  Sample Server Configuration file

[SYSTEM]
Script=radserv.scp
Password-Expiration=90
Password-Warning=7
Password-Minimum=5
BlackList=1
BlackList-Threshold=3
ACE$Challenge-Support=1
ACE$Server-Timeout=45
Log-Failed-Logins=1
Log-Successful-Logins=0
Log-User-Blacklisted=1
Log-User-Password-Change=1
Log-Security-Breach=1

[DICTIONARIES]
1 = /etc/raddb/usrsec.dic
2 = /etc/raddb/dictnary.dat

[SERVER]
Accounting.ip = 192.77.123.156
Accounting.port = 1646

[TABLES]
USERS = users.txt
CLIENTS = clients.txt
DBGROUPS = dbgroups.txt
GROUPS = groups.txt
```

Configuring the Security Data Files

Users File (USERS.TXT)

Each record in USERS.TXT represents a user requiring authentication. User records in the Users file contain user name, password, password expiration and information that the server sends back to the NMC or NETServer client in a response packet.

Before creating user records, you may want to give some thought to the following issues.

- You should have a complete list of user names and, if you intend to set them up as dial back users, the phone numbers that they should be dialed back at (if stored numbers will be used). Make a table of user names that includes passwords and phone numbers.
- Be prepared with all relevant client information that you have set up for your NMC(s) and NETServer(s).
- Consider a strategy for assigning passwords. If you have a few thousand users dialing in to your system, you want a password strategy that is easy for you to use, but still preserves security.
- Consider a strategy for assigning user groups. Assigning users to user groups as they are added simplifies the process. One approach is to create user groups based on the access users need to the DTE. Sample user group names might be MIS, SALES, LAN_ACS, etc.

Configuring the Users File

See Appendix D for a complete description of each attribute/parameter.

1. From an ASCII text editor, open USERS.TXT.
2. Define the main user record attributes:

   Separate the user name and attributes by an equals sign (=). All attributes are on the same line and are separated by commas.

   User-Name: The user name may be up to 16 characters long.

   Mode: If authentication will use Security Dynamics' Security ID, set Mode to 1; otherwise, set to 0.

   Password: The password may be any alphanumeric combination, up to 16 characters.

Password-Expire: This is the date that the user must change his or her password.

User-Group-Name: A user may be a member of only one User Group. Users inherit the characteristics of their user group. Note, however, that user settings for the same attributes override the group settings. For example:

JohnnyB = 0,tuvwxyz,12/31/1995,usr_sales2,

3. Optional. Define response attributes specific to the NMC client.

Dial-In-Security-Mode: This identifies the dial-in security mode used. Valid values are 0 through 2. The default is 0 or Pass-Thru.

0 Pass-Thru: The user is connected directly to the DTE.

1 Dialback Stored Number: The NMC uses a stored number.

2 Dialback Entered Number: The user is prompted to supply a number for dialback.

Dial-In-Call-Restriction: This is a list of restricted / allowed dialback phone numbers issued during dial-in security sessions.

Dial-Out-Call Restriction: This is a list of restricted / allowed dial-out phone numbers.

Failed-Logins: This is the number of failed logins for the current connection.

Request-Dialback-Modem-Selection: This determines whether the NMC prompts dialback users at login for a dialback modem. Valid values are 0 or 1. The default is disabled (0). If enabled (1), the NMC prompts dialback users at login for a dialback modem.

Request-Dialback-Login-Validation: This determines whether the security system issues a login validation sequence during dial back. The default is disabled (0) or no dialback login validation. A value of 1 means that Dialback login validation is enabled.

Dialback-Group-List: This lists dialback groups the user, or user group, is a member of. A user or user group may be a member of zero, one, or more than one dialback groups. Use a semicolon (;) to separate multiple dial groups.

JohnnyB = 0,tuvwxyz,12/31/95,usr_sales2,1,,,,0,,,,,

4. Define response attributes for the NETServer client.

   - If the user is a Login User, you need only set User-Service-Type, Login-Service and Login-Host.

     User-Service-Type: This is the type of service. Valid values are 1 through 4.

1	Login	3	Callback Login (Dialback Login)
2	Framed (Network)	4	Callback Framed (Dialback Network)

     Login-Service: This is the type of connection the user will make with the remote host. Valid values are 0 through 3.

0	Telnet	2	TCP; also called Netdata
1	Rlogin	3	PortMux

     Login-Host: This is the IP address (in dotted notation) of the host that the user will log in to and connect with.

   - If the user is a Network or Framed user, you must configure the remaining attributes.

     Dialback Number: Dialback Login users. This is the AT command string that the client should use to dial the login user back.

     Dialback-Name: Dialback Framed users. This is the name of a host or system in the NETServer's Location Table. Required for Dialback Network users.

     Framed-Filter-ID: This indicates the packet filter that controls the user's access to the network.

     Framed-Address: This is the IP address (in dotted notation) of the remote device.

     Framed-Netmask: This is the netmask (in dotted notation) of the remote device.

     Framed MTU: This is the Maximum Transmission Unit allowed across the serial network connection.

     Framed-IPXNet: This is the IPX network number, in hexadecimal, of the authenticated network user.

     Framed-Protocol: This is the user's protocol. Valid values are 1 (PPP) and 2 (SLIP).

Framed-Routing: This determines whether the serial port permits RIP packets to be sent or received across the serial port interface.

0	None	2	Listen
1	Send	3	Send and Listen

Framed-Compression: This determines the type of compression used. Valid values are 0 (no compression) and 1 (Van Jacobson TCP/IP header compression).

For example, a full Network or Framed user configuration might be as follows:

```
JohnnyB = 0,tuvwxyz,12/31/95,usr_sales2,1,,,,0,,,,,,,3,0,
    192.77.203.66,,,,,,,,
```

5. Save the file

Sample Users File

```
# /usr/tcm/data/security/users.txt -- USR Security Data File
# Table Name: USERS
#    Updated: Fri Oct 13 15:28:42 1995

# Field structure:
[structure]
USER-NAME                                          = string(16)
MODE                                               = string(1)
PASSWORD                                           = string(16)
USER-GROUP-NAME                                    = string(16)
PASSWORD-EXPIRE                                    = string(10)
DIAL-IN-SEC-MODE                                   = string(1)
DIAL-IN-CALL-REST                                  = string(255)
DIAL-OUT-CALL-REST                                 = string(255)
FAILED-LOGINS                                      = numeric
REQ-DB-MDM-SEL                                     = string(1)
REQ-DB-LOGIN-VALID                                 = string(1)
DIALBACK-NAME-LIST                                 = string(255)
DIALBACK-NO                                        = string(255)
DIALBACK-NAME                                      = string(16)
USER-SERVICE-TYPE                                  = string(1)
LOGIN-SERVICE                                      = string(1)
LOGIN-HOST                                         = string(15)
LOGIN-TCP-PORT                                     = string(3)
FRAMED-FILTER-ID                                   = string(18)
FRAMED-ADDRESS                                     = string(15)
FRAMED-NETMASK                                     = string(15)
FRAMED-MTU                                         = string(4)
FRAMED-IPX-NETWORK                                 = string(8)
FRAMED-PROTOCOL                                    = string(1)
FRAMED-ROUTING                                     = string(1)
FRAMED-COMPRESSION                                 = string(1)

# Data:
[data]
JohnnyB =
0,tuvwxyz,12/31/1995,usr_sales2,1,,,,0,,,,,,,3,0,192.77.203.66,,,,,,,,
JLEONG = 1,upt1,upassg0,11/1/1995,,,,3,,,,,,1,0,192.77.203.86,23,,,,,,,
RMURPHY = 1,upt1,upassg0,11/1/1995,,,,0,,,,,,1,0,192.77.203.86,23,,,,,,,
```

Clients Data File (CLIENTS.TXT)

The Clients security data file lists the clients (NMCs and NETServers). The Security Server supports an unlimited number of clients. Each client should be on a separate line. You must also configure the clients for RADIUS authentication. See the NMC and/or NETServer documentation for details.

See Appendix D for a complete description of each parameter.

1. From an ASCII text editor, open CLIENTS.TXT.
2. Define the IP address of the client in dotted notation, and the RADIUS secret used by the client and the Security Server.

   The RADIUS secret can be up to 16 characters long.

   Enclose the IP address and secret in double quotes (") and use an equal sign(=)to separate them. For example:

   "192.77.203.117"="12345678abcdefgh"
3. Save the file

Sample Clients Data File

```
#       CLIENTS.TXT   --  Sample Client Security Data file

[structure]
ip              =  string (15)
secret          =  string (16)

[data]
"192.77.203.117"="12345678abcdefgh"
"192.77.203.66"="testing123"
```

Important Security Note: Both the User's Password and the Client's Secret are originally entered in clear text form. The Server will encrypt these fields automatically the first time they are accessed.

Dialback Groups File (DBGROUPS.TXT)

Each entry in this file indicates what modem channels will be used for dial back. Each entry has a Dialback-Name, IP address of the client, and a list of modem channels (Modems). A user's entry will have the specific Dialback-Name configured in its Dialback-Group-List attribute.

See Appendix D for a complete description of each parameter.

1. From an ASCII text editor, open DBGROUPS.TXT.
2. Define the Name-Client, 32 characters maximum.

   This is the Dialback-Name and the IP address of the client (NMC or NETServer), separated by a colon (:). For example:

   Chicago:192.77.203.76

3. Define the modem channel(s) that are a part of this Dialback group.

   You must use the format 4nnnnnnnnnnnnnnnnn, where n is a single hexadecimal digit (total number of digits should be 17). You must always begin the with a 4, and each digit (n) represents the channels in a chassis slot that may be used for dialback. See the section, *Modem Slot/Channel Table*, for more information. For example:

   Chicago:192.77.203.76 = 40C00000000000000
   OakBrook:192.77.212.122 = 40FFF000000000000
   Sales_East:192.77.207.101 = 40000FFF000000000

4. Save the file.

Sample Dialback Groups File

```
# Table Name: DBGROUPS          dbgroups.txt -- USR Security Data File

# Field structure:
[structure]
Name-Client                                                    = string(32)
Modems                                                         = string(17)

# Data:
[data]
Chicago:192.77.203.76  = 40C00000000000000
OakBrook:192.77.212.122  = 40FFF000000000000
Sales_East:192.77.207.101  = 40000FFF000000000
```

Modem Slot/Channel Table

Each number you use for the n digit corresponds to a specific slot. The hexadecimal number is the binary representation of the modem channels:

b	b	b	b
Channel 4	**Channel 3**	**Channel 2**	**Channel 1**

For example, the binary number 1010 represents channels 2 and 4. The table below provides a listing of all possible slot combinations, their hexadecimal numbers, and binary representations.

Slots Used for Dialback	Hexadecimal Number	Binary
Channel 1	1	0001
Channel 2	2	0010
Channel 1, 2	3	0011
Channel 3	4	0100
Channel 1, 3	5	0101
Channel 2, 3	6	0110
Channel 1, 2, 3	7	0111
Channel 4	8	1000
Channel 1, 4	9	1001
Channel 2, 4	A	1010
Channel 1, 2, 4	B	1011
Channel 3, 4	C	1100
Channel 1, 3, 4	D	1101
Channel 2, 3, 4	E	1110
Channel 1, 2, 3, 4	F	1111

For example:

4080000B004001200

This means that the dialback group for the specified location will use channel 4 in slot 2, channels 1, 2, and 4 in slot 7, channel 3 in slot 10, channel 1 in slot 13, or channel 2 in slot 14 for dialback.

Groups File (GROUPS.TXT)

See Appendix D for a complete description of each attribute / parameter. All group attributes appear on one line and are separated by commas (,).

1. From an ASCII text editor, open GROUPS.TXT.
2. Define the group name, 16 characters maximum. Then configure NMC and / or NETServer specific attributes. See the section earlier in this chapter, *Configuring the Users File*, Steps 2 through 4 for more information. For example:

   usr_sales1 = 0,1,,,0,0,Sales_East,,,,,,,,,,,,,

4. Save the file.

Sample Groups File

```
# Table Name: GROUPS, groups.txt -- USR Security Data File
#    Updated: Fri Aug 11 15:49:49 1995

# Field structure:
[structure]
USER-GROUP-NAME                                 = string(16)
MODE                                            = string(1)
DIAL-IN-SEC-MODE                                = string(1)
DIAL-IN-CALL-REST                               = string(255)
DIAL-OUT-CALL-REST                              = string(255)
REQ-DB-MDM-SEL                                  = string(1)
REQ-DB-LOGIN-VALID                              = string(1)
DIALBACK-NAME-LIST                              = string(255)
DIALBACK-NO                                     = string(255)
DIALBACK-NAME                                   = string(16)
USER-SERVICE-TYPE                               = string(1)
LOGIN-SERVICE                                   = string(1)
LOGIN-HOST                                      = string(15)
LOGIN-TCP-PORT                                  = string(3)
FRAMED-FILTER-ID                                = string(18)
FRAMED-ADDRESS                                  = string(15)
FRAMED-NETMASK                                  = string(15)
FRAMED-MTU                                      = string(4)
FRAMED-IPX-NETWORK                               = string(8)
FRAMED-PROTOCOL                                 = string(1)
FRAMED-ROUTING                                  = string(1)
FRAMED-COMPRESSION                              = string(1)

# Data:
[data]
usr_sales1 = 0,1,,,0,0,Sales_East,,,,,,,,,,,,,
usr_sales2 = 0,2,,,0,0,Chicago,1,,,,,,,,,,,,
usr_sales3 = 0,0,,,0,0,       , ,,,,,,,,,,7fff,,7,9
```

Configuring the Authorization Script

IMPORTANT NOTICE: This script can be modified to alter the authentication process of the RADIUS Security Server. Do not directly alter this file. Copy this script to another name before editing it. Then modify RADSERV.CFG so that it uses the new script.

1. From a text editor, make a copy of the authorization script. The copy is the script that you will edit.
2. Make the required changes to the script. See Appendix E for detailed information on the script language.
3. From a text editor, open the Server Configuration file. Change the script line so that the Security Server reads the edited file instead of the original when it starts up.
4. Save the files.

Sample Authorization Script

```
Access-Request:

EMPTY( Response )                    # Always start empty

# Check if a valid User-Name...
IF( Request.User-Name !InTable User )
        Response.Port-Message = “Invalid Username”    # Set reply message
        # Send Reject response to client
        Respond( DICTIONARY.Request-Type.Access-Reject, Response )
        EXIT                                 # All done
ENDIF

# Check User-Password match
IF(                                  \
        MD5(Request.Authenticator,           \
                Client[Request.Client-ID].Secret, \
                Request.Password) !=             \
        User[Request.User-Name].User-Password \
 )
        Response.Port-Message = “Invalid Password”    # Set reply message
        # Send Reject response to client
        Respond( DICTIONARY.Request-Type.Access-Reject, Response )
        EXIT                                 # All done
ENDIF

# Everything matches, so send Accept response
Respond( DICTIONARY.Request-Type.Access-Accept )
EXIT
```

Configuring the Call Restriction List

Call Restriction can be performed on dial out users, and also on dial in users that are configured for the system to call back. The Call Restriction List may contain up to 20 entries to specify phone numbers that are explicitly allowed or disallowed. Each entry must begin with a plus sign (for an allowed number) or a minus sign (for a disallowed number).

When a user initiates a call, the list is processed. It attempts to match the requested phone number to the entries in the list, from top to bottom. Once a positive or negative match is found, list processing stops and the Call Restriction status is applied to the dial request.

The following table explains the Call Restriction List. For more information on wildcards, see the next section, *Using Wildcards in a Call Restriction List*.

Entry	Meaning
+1617$	All calls within the 617 area code are permitted.
+1312$	All calls within the 312 area code are permitted.
-1$	No other long distance calls (outside of 617 and 312) are permitted.
-5551234$	Local number 5551234 is not permitted. The ending dollar sign ensures the user does not try to break the match by adding extra digits. This kind of restriction is useful to disallow calls by some user groups to specified resources, while allowing access to other groups with a different template.
+$	All other local calls are permitted.

Using Wildcards in a Call Restriction List

Wildcards simplify list entries. The single character wildcard is a question mark (?); the global character wildcard (to replace any number of characters) is a dollar sign ($).

We recommend that you always end your list with either +$ or -$, to indicate how to handle numbers not specified in the list. You should also add the $ wildcard after numbers you want to specifically disallow; otherwise, a user could get around the restriction by dialing more numbers to break the exact match.

Appendix A
Standard RADIUS Attributes

Packet Type	Code	Attributes	Description
Access-Request	1	User-Name or NAS-IP-Address, User-Password or CHAP-Password; Others as needed.	Access-Request packets convey information that determines if a user is allowed access to a specific host, and any special services requested for that user. When a password is present, it is hidden using a method based on the RSA Message Digest Algorithm MD5 [3].
Access-Accept	2	Variable	Access-Accept packets provide the configuration information required to begin delivery of services to the user.
Access-Reject	3	None	This packet is sent if the RADIUS server could not authenticate the attributes contained in the Access-Request or Access-Challenge packet.
Accounting-Request	4		
Accounting-Response	5		
Access-Challenge	11	Packet to client: Reply-Message or State. Packet to server: new request ID, User-Password, and State	The RADIUS sends this packet to the NMC/NETServer client. The client prompts the user for a new password, then resubmits its original Access-Request with a new request-ID with the User-Password Attribute replaced by the user's response (encrypted), and including the State Attribute from the Access-Challenge, if any. Only 0 or 1 instances of the State Attribute can be present in a request.

Attribute	Identifier	Type/Value	Description
User-Name	1	8 character string	This attribute contains the alphanumeric name of the user the RADIUS server will authenticate.
User-Password	2	32 char. string	This contains the alphanumeric password associated with the User-Name above.
CHAP-Password	3	16 character string	This is the password provided by a PPP Challenge-Handshake Authentication Protocol (CHAP) user, and is only used in Access-Request packets.
NAS-IP-Address	4	IP address in dotted decimal	This is the IP address of the client. Either this attribute or NAS-Identifier must be present in the Access-Request packet.
NAS-Port	5	integer between 0 and 65535	This is the port number of the NETServer or modem slot of the client that is authenticating the user. Used only in Access-Request packets.
Service Type	6	value	This indicates what type of user is being authenticated. Note that the NETServer refers to Framed Users as Network Users, and Callback Users as Dialback Users. Types are listed below: 1 Login 2 Framed 3 Callback Login 4 Callback Framed 5 Outbound User 6 Administrative User 7 Exec User 8 Authenticate Only
Framed-Protocol	7	value	This indicates what protocol a Framed/Network User uses for the connection. 1 PPP 2 SLIP
Framed-IP-Address	8	IP address in dotted decimal	This is the IP address the user will use for the duration of the connection. Used only in Access-Accept packets.
Framed-IP-Netmask	9	IP address in dotted decimal	This is the IP netmask. Required only if subnetting is used on the network the user will access. Used only in Access-Accept packets.
Framed-Routing	10	value	This indicates what type of RIP messaging is used. Note that the NETServer uses the term Broadcast instead of Send. Used only in Access-Accept packets. 0 None 1 Send routing packets to the network 2 Listen for routing packets from the network 3 Send and Listen

Attribute	Identifier	Type/Value	Description
Filter-ID	11	255 character string	This indicates the packet filter that control the user's access to the network.
Framed-MTU	12	integer between 64 and 65535	This is the Maximum Transmission Unit used for the connection. 1500 is the PPP default. 1060 is the SLIP default.
Framed-Compression	13	value of 1 or 2	This attribute indicates that compression will be used. 1 Van Jacobson TCP/IP header compression 2 IPX header compression
Login-IP-Host	14	IP address in dotted decimal form, 0xFFFFFFFF, or 0	This indicates what host a Login or Callback Login user will be connected with after authentication. Used only in Access-Accept packets. If set to 0xFFFFFFFF, the user determines what host to connect to. The NETServer port's Host Type should be set to prompt, or the user's entry in the NETServer's User Table should have Host set to prompt. If set to 0, the NMC/NETServer select the host. The NETServer port's Host Type to should be set to default or specified, or the user's entry in the NETServer's User Table should have Host set to default or specified.
Login-Service	15	value between 0 and 4	This determines what network service is used for the connection. Used only in Access-Accept packets. 0 Telnet 1 Rlogin 2 TCP; also called Netdata 3 PortMux 4 LAT
Login-Port	16	integer between 64 and 65535	This is the login service port. Unless users are configured for TCP/Netdata, this attribute is optional. Used on in Access-Accept packets.
Unassigned	17	—	Attribute has not been assigned.
Reply-Message	18	255 character string	This is a text message the users see. It may accompany a Access-Request, Access-Reject, or Access-Challenge. Multiple Reply-Messages may be included. ASCII characters only.
Login-Callback-Number	19	255 character string	This is the AT command dialing string the client (NMC or NETServer) uses to dial back the user. Used in both Access-Request and Access-Accept packets.

Attribute	Identifier	Type/Value	Description
Framed-Callback-ID	20	255 character string	This is the location that the NMC/NETServer will use to dial the Framed/Network user back.
Unassigned	21	—	Attribute has not been assigned.
Framed-Route	22	IP address(es) in dotted decimal with metric	This indicates the destination address (optional), IP address of the default router/gateway and the metric.
Framed-IPX-Network	23	IPX network number; 8 digit hexadecimal	This is the user's IPX network number. Used in Access-Accept packets. If set to 0xFFFFFFFF, the client sets the IPX network number. For example, the NETServer might use the IPX network number from the user's entry in its User Table.
State	24	application specific	This is an optional attribute sent by the RADIUS server to the client in an Access-Challenge, and returned unchanged to server by the client. See Appendix B, *USR Specific Attributes*, for more information.
Class	25	application specific	This attribute is sent by the RADIUS server to the client in an Access-Accept. The client sends this attribute, unchanged, to the accounting server as a part of its Accounting-Request packet.
Vendor-Specific	26	application specific	See Appendix B for details on U.S. Robotics' extensions to RADIUS.
Session-Timeout	27	32-bit unsigned integer	This is the maximum amount of time in seconds that the user may be connected.
Idle-Timeout	28	32-bit unsigned integer	This is the maximum amount of time, in seconds that the connection may remain idle before the client disconnects.
Termination-Action	29	value of 0 or 1	This attribute indicates the client's response when the user disconnects, the Session-Timeout value is reached, or the Idle-Timeout is reached. 0 Default; the client's default response to disconnect 1 RADIUS-Request; the client may send a new Access-Request to the server
Client-Port-DNIS	30	application specific	This attribute contains the phone number the user's call came in on (the phone number the user called in to).
Caller-ID	31	application specific	This attribute contains the phone number that the user called in from. Used in Access-Request packets only.

Attribute	Identifier	Type/Value	Description
NAS-Identifier	32	255 character string	This attribute identifies the client that originated the Access-Request packet. Either this attribute or NAS-IP-Address must be present in the Access-Request packet.
Proxy State	33	application specific	This attribute is sent by a proxy RADIUS server to another RADIUS server when forwarding an Accept-Request. It is returned unmodified in the Access-Accept, Access-Reject, or Access-Challenge packets. The proxy RADIUS server removes this attribute from Access-Accept, Access-Reject or Access-Challenge packet before forwarding the packet the client.
Login-LAT-Service	34	application specific	This indicates what LAT service the user is connected to. LAT (Local Area Transport) is a protocol that provides an efficient means of logically connecting terminal servers to one or more nodes on the same local area network (LAN).
Login-LAT-Node	35	application specific	This indicates the node that users will be connected to. Used only when LAT is specified as the Login-Service
Login-LAT-Group	36	256 bit bitmap	This identifies the LAT group codes the user is authorized for. Used only in Access-Request packets when LAT is specified as the Login-Service. One or more group codes may be assigned.
Framed-AppleTalk-Link	37	0 through 65535; network number	This indicates the AppleTalk network number that is used for the serial link when the user is another router. Not used if the user is not a router. Used only in Access-Accept packets. A value of zero (0) means that the serial link is unnumbered.
Framed-AppleTalk-Network	38	0 through 65535; network number	This indicates the AppleTalk network number that the NMC/NETServer client should probe to allocate an address for the user. Used only if the user is not a router. Used only in Access-Accept packets. Multiple instances are permitted.
Framed-AppleTalk-Zone	39	application specific	This indicates the AppleTalk Default Zone that applies to the user. Used only in Access-Accept packets. Multiple instances are permitted.
Acct-Status-Type	40	value of 1 or 2	This indicates whether the Accounting-Request marks the beginning or end of the user's service/connection. 1 Start 2 Stop

Attribute	Identifier	Type/Value	Description
Acct-Delay-Time	41	integer	This indicates how many seconds the client has been trying to send a record to the RADIUS Server. Subtracting this value from the time of arrival lets you find out approximately when the Accounting-Request packet was generated.
Acct-Input-Octets	42	integer	This indicates how many octets have been received since the connection began. Used only in Accounting-Request packets when Acct-Status-Type is set to Stop (2).
Acct-Output	43	integer	This indicates how many octets have been sent since the connection began. Used only in Accounting-Request packets when Acct-Status-Type is set to Stop (2).
Acct-Session-Id	44	255 character string	This is a unique Accounting ID and is used to make it easy to match start and stop records in a log file. Both start and stop records must have the same Acct-Session-Id.
Acct-Authentic	45	value of 1 or 2	This indicates how the user was authenticated. Users who are connected without being authenticated should not generate Accounting records. 1 RADIUS 2 Local
Acct-Session-Time	46	integer	This indicates how many seconds the user has been connected. Used only in Accounting-Request packets when Acct-Status-Type is set to Stop (2).
Acct-Input-Packets	47	integer	This indicates how many packets have been received from the port since the connection began. Applies only to a Frame User. Used only in Accounting-Request packets when Acct-Status-Type is set to Stop (2). Note that the NETServer refers to Framed Users as Network Users.
Acct-Output-Packets	48	integer	This indicates how many packets have been sent to the port since the connection began. Applies only to a Frame User. Used only in Accounting-Request packets when Acct-Status-Type is set to Stop (2). Note that the NETServer refers to Framed Users as Network Users.

Appendix B
USR RADIUS Extensions

Attribute Name	Identifier	Type/Value	Description
Allowed-Dial-Back-Modems (233)	233	string, 255 characters max.	Applies only for dialback groups and to both Dial-Back Stored and Dial-Back Entered Number modes. This is a string field that defines what modems may dial back a user. Modems are specified in slot/ channel or dial back modem group format in a comma separated list. The default is empty. You may specify a range of modems. If you use SAME_MODEM, then the modem used for dial-in will be the dialback modem. All modems must be in the same chassis. When alternate modems are selected for dialback, the system checks the specified dialback modem for availability and reserves it. If the modem is in use, the system advises the user and terminates the call.
Dial-In-Call-Restriction	228	string, 255 characters max.	This field represents a list of restricted/allowed dial back phone numbers issued during dial-in security sessions.
Dial-In-Security-Mode	224	value	This field identifies the type of dial-in security mode used for this user once the initial login sequence is passed. It specifies one of three possible values: Pass-Thru (value=0), where the user is connected directly to the DTE. Dial-Back Stored Number (value=1), where the NMC disconnects the modem and dials back a stored number for this user. Dial-Back Entered Number (value=2), where the user is prompted to supply a number at which to be called back. The default is Pass-Thru.
Dial-Out-Call Restriction	229	string, 255 characters max.	This field represents a list of restricted/allowed dial-out phone numbers.
Dialback-Group-Membership	227	string, 255 characters max.	This field represents a list of dialback groups that this user, or user group, is a member of. A user or user group may be a member of zero, one, or more dialback groups.

Attribute Name	Identifier	Type/Value	Description
Request-Dial-Back-Login-Validation	226	integer	This is a toggle field. If enabled, the security system issues a login validation sequence during dial back. The default is not to perform dial-back login validation.
Request-Dial-Back-Modem-Selection	225	integer	This is a toggle field. If enabled, the NMC prompts dial-back users at login for a dial-back modem. The system displays all valid choices from the list of Allowed Dial-Back Modems in a menu. Users must enter the number that appears on the menu next to their choice. If this field is disabled, the NMC checks the Allowed Dial-Back modem field, polls the modems, and reserves the first available modem to dial back. The default for this field is disabled.
State		4 character string	Indicates the challenge Service-type and a State used by the specific Challenge-Response process. Encoding is proprietary to the USR Security Server.

Other Attributes and Data Types

		RADIUS Attributes (Value) and Data Types									
USR Event	**Event ID**	**User Name (1)**	**Client ID (4)**	**Client Port ID (5)**	**Dial Back No. (19)**	**Vendor Spec. (26)**	**Acct. Status Type (40)**	**Acct. Input Octets (42)**	**Acct. Output Octets (43)**	**Acct. Sess. ID (44)**	**Acct. Sess. Time (46)**
Card Inserted	06		S,A	S,I		S,V				S,I	
Card Removed	07		S,A	S,I		S,V				S,I	
Power Supply Warning	08		S,A	S,I		S,V				S,I	
Power supply failed	09		S,A	S,I		S,V				S,I	
Temperature Warning	10		S,A	S,I		S,V				S,I	
Fan Failed	11		S,A	S,I		S,V				S,I	
Watchdog Timeout	12		S,A	S,I		S,V				S,I	
Management Bus Failure	13		S,A	S,I		S,V				S,I	
Incoming Connection Est.	14		S,A	S,I		S,V	S,I			S,I	
Outgoing Connection Est.	15		S,A	S,I		S,V	S,I			S,I	
Incoming Connection Term.	16	S,C	S,A	S,I	S,C	S,V	S,I	O2,I	O2,I	S,I	S,I
Outgoing Connection Term.	17	S,C	S,A	S,I		S,V	S,I	O2,I	O2,I	S,I	S,I
Connection Attempt Failure	18		S,A	S,I		S,V				S,I	
Connection Timer Expired	19		S,A	S,I		S,V				S,I	
DTE Transmit Data Idle	20		S,A	S,I		S,V				S,I	
DTR True	21		S,A	S,I		S,V				S,I	
DTR False	22		S,A	S,I		S,V				S,I	
BLER Count at Threshold	23		S,A	S,I		S,V				S,I	
Fallback Count at Threshold	24		S,A	S,I		S,V				S,I	
No Dial Tone	25		S,A	S,I		S,V				S,I	
No Loop Current	26		S,A	S,I		S,V				S,I	
Modem Reset By DTE	32		S,A	S,I		S,V				S,I	
Modem Ring No Answer	33		S,A	S,I		S,V				S,I	
DTE Ring No Answer	34		S,A	S,I		S,V				S,I	
Dial Out Login Failure	42	S,C	S,A	S,I		S,V				S,I	
Dial In Login Failure	43	S,C	S,A	S,I		S,V				S,I	
Dial Out Restricted No.	44	S,C	S,A	S,I		S,V				S,I	
Dial Back Restricted No.	45	S,C	S,A	S,I		S,V				S,I	
User Blacklisted	46	S,C	S,A	S,I		S,V				S,I	
Blacklisted User Login Att.	47	S,C	S,A	S,I		S,V				S,I	
Resp. Att. Limit Exceeded	48	S,C	S,A	S,I		S,V				S,I	
Login Att. Limit Exceeded	49	S,C	S,A	S,I		S,V				S,I	

S = Standard
O = Optional. Part of Call Termination statistics group n.
Data Types: A = IP Address, I = Integer, C = Character String, T = GMT date/time,
V = Variable (see Vendor Specific attribute table).

Table 1. Log Message RADIUS Attributes and Data Types

			USR Event IDs															
USR Attribute	**Attr. Value**	**Data Type**	**06**	**07**	**08**	**09**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**
Actual-Voltage	0xBF32	I			S													
Back-Channel-Data-Rate	0x007C	I											O4	O4				
Block-Error-Count-Limit	0x00C0	I																
Blocks-Received	0x0076	I											O2	O2				
Blocks-Resent	0x0077	I											O2	O2				
Blocks-Sent	0x0075	I											O2	O2				
Call-End-Date-Time	0xBFF6	T											S	S				
Call-Start-Date-Time	0xBFF7	T											S	S				
Card-Type	0xBE85	I	S	S														
Channel	0xBF38	I							S	S	S	S	S	S	S	S	S	S
Characters-Received	0x0072	I											O2	O2				
Characters-Sent	0x0071	I											O2	O2				
Chassis-Slot	0xBF39	I	S	S					S	S	S	S	S	S	S	S	S	S
Chassis-Temp-Threshold	0xBE84	I					S											
Chassis-Temperature	0xBF31	I					S											
Connect-Term-Reason	0x009B	I											S	S				
Connect-Time-Limit	0xBFE7	I														S		
Default-DTE-Data-Rate	0x005E	I											O4	O4				
DTE-Data-Idle-Timout	0x0048	I															S	
DTE-Ring-No-Answer-Limit	0xBF17	I																
DTR-False-Timeout	0x00BE	I																
DTR-True-Timeout	0x00DA	I																S
Equalization-Type	0x006F	I											O4	O4				
Event-Date-Time	0xBF2F	T	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S
Event-Id	0xBFBE	I	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S
Expected-Voltage	0xBF33	I			S													
Failure-to-Connect-Reason	0x0069	I													S			
Fallback-Enabled	0x0070	I											O4	O4				
Fallback-Limit	0x00BF	I																
Final-Rx-Link-Data-Rate	0xBF2C	I											O3	O3				
Final-Tx-Link-Data-Rate	0x006B	I											O3	O3				
Initial-Rx-Link-Data-Rate	0xBF2D	I											O3	O3				
Initial-Tx-Link-Data-Rate	0x006A	I											O3	O3				
Last-Callers-Number-ANI	0x00E9	C											S					
Last-Number-Dialed-In-DNIS	0x00E8	C											S					
Last-Number-Dialed-Out	0x0066	C												S				
Line-Reversals	0x007A	I											O2	O2				
Modulation-Type	0x006C	I											O4	O4				

On = Optional. Part of Call Termination statistics group n.
S = Standard
Data Types: C = Character String, I = Integer, T = GMT date/time

Table 2. USR Vendor Specific Attributes and Data Types per Event Type (Event IDs 06 through 21)

			USR Event IDs															
USR Attribute	**Attr. Value**	**Data Type**	**06**	**07**	**08**	**09**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**
Number-of-Blers	0x007D	I											O3	O3				
Number-Of-Characters-Lost	0x007B	I											O2	O2				
Number-of-Fallbacks	0x007F	I											O3	O3				
Number-of-Link-NAKs	0x0081	I											O3	O3				
Number-of-Link-Timeouts	0x007E	I											O3	O3				
Number-of-Rings-Limit	0xBFE6	I																
Number-of-Upshifts	0x0080	I											O3	O3				
Originate-Answer-Mode	0x0068	I											O4	O4				
Power-Supply-Number	0xBF34	I				S												
Retrains-Granted	0x0079	I											O3	O3				
Retrains-Requested	0x0078	I											O3	O3				
Security-Login-Limit	0xBEDE	I																
Security-Resp-Limit	0xBEFA	I																
Simplified-MNP-Levels	0x0099	I											O4	O4				
Simplified-V42bis-Usage	0x00C7	I											O4	O4				
Sync-Async-Mode	0x0067	I											O4	O4				

On = Optional. Part of Call Termination statistics group n.
S = Standard
Data Types: C = Character String, I = Integer, T = GMT date/time

Table 2. (Continued)

			USR Event IDs															
USR Attribute	**Attr. Value**	**Data Type**	**22**	**23**	**24**	**25**	**26**	**32**	**33**	**34**	**42**	**43**	**44**	**45**	**46**	**47**	**48**	**49**
Actual-Voltage	0xBF32	I																
Back-Channel-Data-Rate	0x007C	I																
Block-Error-Count-Limit	0x00C0	I		S														
Blocks-Received	0x0076	I																
Blocks-Resent	0x0077	I																
Blocks-Sent	0x0075	I																
Call-End-Date-Time	0xBFF6	T																
Call-Start-Date-Time	0xBFF7	T																
Card-Type	0xBE85	I																
Channel	0xBF38	I	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S
Characters-Received	0x0072	I																
Characters-Sent	0x0071	I																
Chassis-Slot	0xBF39	I	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S
Chassis-Temp-Threshold	0xBE84	I																
Chassis-Temperature	0xBF31	I																
Connect-Term-Reason	0x009B	I																
Connect-Time-Limit	0xBFE7	I																
Default-DTE-Data-Rate	0x005E	I																
DTE-Data-Idle-Timout	0x0048	I																
DTE-Ring-No-Answer-Limit	0xBF17	I							S	S								
DTR-False-Timeout	0x00BE	I	S															
DTR-True-Timeout	0x00DA	I																
Equalization-Type	0x006F	I																
Event-Date-Time	0xBF2F	T	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S
Event-Id	0xBFBE	I	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S
Expected-Voltage	0xBF33	I																
Failure-to-Connect-Reason	0x0069	I																
Fallback-Enabled	0x0070	I																
Fallback-Limit	0x00BF	I			S													
Final-Rx-Link-Data-Rate	0xBF2C	I																
Final-Tx-Link-Data-Rate	0x006B	I																
Initial-Rx-Link-Data-Rate	0xBF2D	I																
Initial-Tx-Link-Data-Rate	0x006A	I																
Last-Callers-Number-ANI	0x00E9	C																
Last-Number-Dialed-In-DNIS	0x00E8	C																
Last-Number-Dialed-Out	0x0066	C																
Line-Reversals	0x007A	I																
Modulation-Type	0x006C	I																

On = Optional. Part of Call Termination statistics group n.
S = Standard
Data Types: C = Character String, I = Integer, T = GMT date/time

Table 3. USR Vendor Specific Attributes and Data Types per Event Type (Event IDs 22 through 49)

			USR Event IDs															
USR Attribute	**Attr. Value**	**Data Type**	**22**	**23**	**24**	**25**	**26**	**32**	**33**	**34**	**42**	**43**	**44**	**45**	**46**	**47**	**48**	**49**
Number-of-Blers	0x007D	I																
Number-Of-Characters-Lost	0x007B	I																
Number-of-Fallbacks	0x007F	I																
Number-of-Link-NAKs	0x0081	I																
Number-of-Link-Timeouts	0x007E	I																
Number-of-Rings-Limit	0xBFE6	I																
Number-of-Upshifts	0x0080	I																
Originate-Answer-Mode	0x0068	I																
Power-Supply-Number	0xBF34	I																
Retrains-Granted	0x0079	I																
Retrains-Requested	0x0078	I																
Security-Login-Limit	0xBEDE	I																S
Security-Resp-Limit	0xBEFA	I															S	
Simplified-MNP-Levels	0x0099	I																
Simplified-V42bis-Usage	0x00C7	I																
Sync-Async-Mode	0x0067	I																

On = Optional. Part of Call Termination statistics group n.
S = Standard
Data Types: C = Character String, I = Integer, T = GMT date/time

Table 3. (Continued

Appendix C
Accounting Server Files and Records

OUTDEF.DAT File

The following section describes the syntax required to program the OUTDEF.DAT file.

Keywords

Keyword	Description
FILE	Which log file is to be created (should always be followed by a list of attributes to be included in the file)
TIME	Whether local or GMT time is to be used
PATH	Where the log file is to be stored
DISK_LOW	What action to take on low disk space
DISK_FULL	What action to take on full disk space
NEW_LOG_FILE	What action to take when log is closed

FILE

This indicates a log file to be defined, and has three fields:

- keyword FILE
- reference name of the log file (e.g., NMC_CONNECTIVITY)
- suffix of the log file (3 character limit), used to indicate the type of information stored (e.g., con, rad, or nmc)

The name of the log file is the date in the format yymmdd followed by "." and the suffix (con, nmc, or rad).

Example:

FILE NMC_CONNECTIVITY con

TIME

This indicates what time zone to use for any time that appears in the log file. It has two fields:

- keyword TIME
- time zone (must be GMT or LOCAL)

If TIME does not appear, the default is LOCAL.

Example:

TIME GMT

PATH

This indicates the directory that the log files are written to. It has two fields:

- keyword PATH
- the directory path

If PATH does not appear, the default is the directory in which the accounting server executes, represented as a period.

Example:

PATH .\files

NOTE: The directory must exist in order for the log file to be created. The example specifies a subdirectory (indicated by the "\") named "files" that is under the directory from which the accounting server is launched (indicated by the "."). If the server were to be launched from a different directory (e.g., using Program Manager), the path indicated in the example may be invalid and the file may not be created.

DISK_LOW

- This keyword indicates what to do when the free disk space is at a minimum. It has three fields:
- keyword DISK_LOW
- .exe/.bat file to be executed (optional)
- free disk space threshold in MBytes (optional)

Disk space is checked periodically, and if the free disk space threshold is reached, the defined .exe / .bat file is executed using the PATH of the log files as a parameter. This should be used to free up disk space, typically by purging the accounting log files. If the DISK_LOW keyword does not appear, a default of 10 Mbytes is assigned for the free disk space threshold.

Example:

```
DISK_LOW                disklow.bat          12
```

DISK_FULL

This keyword indicates what to do when the free disk space is at a fatal level. It has three fields:

- keyword DISK_FULL
- .exe / .bat file to be executed (optional)
- fatal free disk space threshold in MBytes (optional)

When the disk full threshold is reached, the defined .exe / .bat file is executed. This .exe / .bat may be used in the event that DISK_LOW fails to increase the free disk space; typically, this might be used to initiate an alarm of some kind, page an administrator, etc. If DISK_FULL does not appear in the file, a default of 5 MBytes is assigned for the fatal free disk space limit. If the .exe / .bat is omitted, the file USRBEEP.EXE is run to provide alarm indication. After the file is executed the server application will terminate.

Example:

```
DISK_FULL               diskfull.bat         6
```

NEW_LOG_FILE

This keyword indicates what to do when log files are closed at midnight. It has two fields:

- keyword NEW_LOG_FILE
- .exe / .bat file to be executed

At midnight, each log file is closed and new log files reflecting the new date are created. After the new log files are created, the .exe / .bat is executed using the PATH of the log files as a parameter. This .exe / .bat typically purges and / or imports the old log file into a database.

Example:

```
NEW_LOG_FILE            newlog.bat
```

Format Specifiers

Each attribute definition has two fields:

- attribute name
- format specifier (optional)

The format specifier overrides the default format specified by the dictionary file. Each attribute has a type with a number of format options.

All transactions are composed of Attribute/Value Pairs. The value of each attribute is specified as one of 4 data types. Enumerated values are stored in the user file with dictionary VALUE translations for easy administration. Valid data types are:

Data Type	Definition	Format Options
string	0–254 octets	no additional formatting available
ipaddr	4 octets in network byte order	Three different format options are available: IPdot, e.g., 192.203.77.91 IPhex, e.g., C0CB4D5B IPdec, e.g., 3234549083
integer	32 bit value in big endian order (high byte first)	Two different format options are available: [Text]% % replaced by the integer value value print out the VALUE dictionary text
date	32 bit value in big endian order — seconds since 00:00:00 GMT, Jan. 1, 1970	Five different format options are available: date1 mm/dd/yy hh:mm:ss date2 dd-mm-yy hh:mm:ss date3 yymmddhhmmss date4 example: Wed Jan 02 02:03:55 1980 date5 seconds since 00:00:00 Jan. 1, 1970

Log File Format Examples

After the keyword FILE, which begins defining an output file, you must list attributes whose values you want to be included in the file. Each attribute is represented by a column in the file. Beside each entry, an optional format string can be provided that will override the format string provided by the dictionary.

Example—Connection Log

These records provide call information from the NMC client, using two types of events: Event-Id = Incoming-Connection-Terminated or Event-Id = Outgoing-Connection-Terminated. The example below contains only selected attributes (many more are available).

```
FILE    NMC_CONNECTIVITY            .con
        Client-Id                   IPdot
        Chassis-Slot                %
        Channel                     %
        Event-Date-Time             Date1
        Event-Id                    %
        Acct-Session-Id
        User-Name
        Call-Start-Date-Time        Date1
        Call-End-Date-Time          Date1
        Connect-Term-Reason         %
        Acct-Session-Time           %
        Acct-Input-Octets           %
        Acct-Output-Octets          %
        Default-DTE-Data-Rate       %
        Last-Number-Dialed-In-DNIS
        Last-Number-Dialed-Out
        Last-Callers-Number-ANI
```

Example—Event Log

The event log table stores event information generated by an NMC client, but not recorded in the Call Termination Log. This includes such things as Module-Inserted, Connection-Attempt-Failed, etc.

```
FILE    NMC_EVENTLOG                .nmc
        Client-Id                   IPdot
        Chassis-Slot                %
        Channel                     %
        Event-Date-Time             Date1
        Event-Id                    %
        Acct-Session-Id
        Failure-to-Connect-Reason   %
        Server-Time                 Date1
        Client-Port-Id              %
```

Example—Native RADIUS Log

The generic RADIUS table stores events that do not have the USR specific Event-Id attribute (i.e., non-NMC events). Any attributes of interest must be entered into this file definition or they will be discarded.

```
FILE    GENERIC_RADIUS          .rad
        Client-Id               IPdot
        Server-Time             Date1
        Acct-Status-Type        %
        Acct-Session-Id
        Acct-Session-Time       %
```

Accounting Server Log Records

The NMC client can receive event records from 65 possible sources—one NMC and a possible 64 modems (16 Quad Modem cards). The NMC client also has an input queue capable of handling 65 event records. This allows for the maximum possible requesters to be issuing event records simultaneously. If the input queue of the NMC client becomes overloaded, the requester's attempt to issue an event record will be rejected. It is up to the requester to handle these rejections. Until the rejections are dealt with, the events are discarded.

The NMC client removes an event record from its input queue, formats it properly for the Accounting server, sends it, and waits for the reply before starting on the next event record. If communication is lost between the NMC and the primary server, data will be sent to the secondary server. If communication is lost with the secondary server before it is re-established with the primary server, the NMC client discards the event records.

Event Log

The event record format may be extended to include the necessary event information. The following information is included with all event records.

- IP Address of Chassis
- UDP Port Number
- Event Sequence Number
- Event Identifier
- Event Date/Time Stamp

The following table identifies the information sent in the log record for each event, in addition to the standard fields listed above. The Generic Slot Record consists of the Slot Number and Module Type. The Entity Index indicates a description beyond slot, e.g., modem channel.

Events	Additional Information In Event Record
Module Inserted (6)	Generic Slot Record
Module Removed (7)	Generic Slot Record
PSU Voltage out of Range (8)	Expected voltage Actual voltage
PSU Failed (9)	PSU Index
HUB Temperature out of Range (10)	Chassis temperature Temperature threshold
Fan Failed (11)	No additional information
Module Watchdog Time-out (12)	Generic Slot Record
Management Bus Failure (13)	Generic Slot Record Entity Index
Incoming Connection Established (14)	Generic Slot Record Call status Entity Index
Outgoing Connection Established (15)	Generic Slot Record Call status Entity Index
Incoming Connection Terminated (16)	See following description
Outgoing Connection Terminated (17)	See following description
Connection Attempt Failed (18)	Generic Slot Record Entity Index Failure to connect reason
Connection Time Limit Expired (19)	Generic Slot Record Entity Index Connect time limit
DTE Transmit Idle (20)	Generic Slot Record Entity Index DTE Data Idle Timeout
DTR True (21)	Generic Slot Record Entity Index DTR True Timeout
DTR False (22)	Generic Slot Record Entity Index DTR False Timeout
Block Error Count at Threshold (23)	Generic Slot Record Entity Index Block Error Count Limit
Fallback Count at Threshold (24)	Generic Slot Record Entity Index Fallback Count Limit
No Dial Tone Detected (25)	Generic Slot Record Entity Index
No Loop Current Detected (26)	Generic Slot Record Entity Index
Modem Reset by DTE (32)	Generic Slot Record Entity Index
Modem Ring No Answer (33)	Generic Slot Record Entity Index Ring No Answer Limit
DTE Ring No Answer (34)	Generic Slot Record Entity Index Ring No Answer Limit
Dial Out Login Failure (42)	Generic Slot Record Entity Index User Name

Events	Additional Information In Event Record
Dial In Login Failure (43)	Generic Slot Record Entity Index User Name
Dial Out Restricted Number (44)	Generic Slot Record Entity Index User Name
Dial Back Restricted Number (45)	Generic Slot Record Entity Index User Name
User Blacklisted (46)	Generic Slot Record Entity Index User Name
Attempted Login by Blacklisted User (47)	Generic Slot Record Entity Index User Name
Response Attempt Limit Exceeded (48)	Generic Slot Record Entity Index User Name Security Response Limit
Modem Login Attempt Limit Exceeded (49)	Generic Slot Record Entity Index User Name Security Login Limit

Call Termination Log

Event Records for Incoming Connection Terminated (16) and Outgoing Connection Terminated (17) are identical, except as noted in the table below (ANI, Number Dialed, and DNIS in Group 1).

Incoming Connection Terminated (16) and Outgoing Connection Terminated (17)	Generic Slot Record Entity Index **Usage Statistics (Group 1 - always sent):** User Name (mdmCsSecurityUserName) Call Start Date/Time Call End Date/Time Call Termination Reason (mdmCsDisconnectReason) ANI (mdmCsLastCallingPartyNum)--INCOMING ONLY Number Dialed (mdmCsLastNumberDialedOut)--OUTGOING ONLY DNIS (mdmCsLastNumberDialedIn)--INCOMING ONLY Call Duration (mdmCsCallDuration) **Data Transfer Statistics (Group 2):** Characters Sent (mdmCsCharsSent) Characters Received (mdmCsCharsReceived) Octets Sent (mdmCsOctetsSent) Octets Received (mdmCsOctetsReceived) Blocks Sent (mdmCsBlocksSent) Blocks Received (mdmCsBlocksReceived) Blocks Resent (mdmCsBlocksResent) Characters Lost (mdmCsCharsLost) Line Reversals (mdmCsLineReversalQty) **Performance Statistics (Group 3):** Block CRC Errors (mdmCsBlerQty) Link NAKs (mdmCsLinkNakQty) Link Fallbacks (mdmCsFallbackQty) Link Upshifts (mdmCsUpshiftQty) Link Timeouts (mdmCsLinkTimeoutQty) Initial Link TX Rate (mdmCsInitialTxLinkRate) Final Link TX Rate (mdmCsFinalTxLinkRate) Initial Link RX Rate (mdmCsInitialRxLinkRate) Final Link RX Rate (mdmCsFinalRxLinkRate) Retrains Requested (mdmCsRetrainsRequested) Retrains Granted (mdmCsRetrainsGranted) **Operating Mode Statistics (Group 4):** Sync/Async Mode (mdmCsSyncAsyncModeUsed) Modulation Type (mdmCsModulationType) Originate/Answer Mode (mdmCsOriginateAnswer) Error Control Type (mdmCsErrorControlType) Data Compression Type (mdmCsCompressionType) HST Back Channel Rate (mdmCsBackChannelRate) Default DTE Data Rate High Freq Equalization (mdmCsEqualizationType) On-Line Fallback (mdmCsFallbackEnabled)

Appendix D
Security Data Files

Server Configuration File

The Server Configuration file (RADSERV.CFG) is located in the same directory as the server. It is an ASCII file containing the following information:

- The location of all Dictionary files
- The location of all security files used in the authentication process
- How to configure the RADIUS server(s)

Parameter	Description
[SYSTEM]	This indicates the beginning of the SYSTEM parameters.
SCRIPT=full-path-name	This names the main authentication script file, and includes the device and directory name.
Password-Expiration=days	This sets how long in days, that a particular password remains valid before it must be changed. A value of 0 means that passwords have no expiration time limit.
Password-Warning=days	This sets the number of days before expiration a user is warned of the upcoming expiration date. A warning message is displayed to users. Only used when Password-Expiration is set to any value other than 0.
Password-Minimum=n	The minimum number of characters required for a user's password.
BlackList-Threshold=n	This is the number of consecutive login failures the server permits before blacklisting a user.
BlackList=0|1	This indicates whether the RADIUS server is configured for Black-List processing. 0=Disabled or Off and 1=Enabled or On.
ACE$Challenge_Support=0|1	This indicates whether the NMC and/or NETServer client supports a RADIUS challenge. 0=Challenge Disabled or Off and 1=Challenge Enabled or On.
ACE$Server-Timeout=seconds	This specifies the amount of time in seconds that the Security Server will wait for a response from the Security Dynamics ACE/Server before declaring the server unreachable.

Parameter	Description
Log-Failed-Logins=0|1	Indicates whether a failed login attempt by a user will be logged to the USR Accounting Server log file(s). 0=Event Not Logged and 1=Event is Logged.
Log-Successful-Logins=0|1	Indicates whether a successful login attempt by a user will be logged to the USR Accounting Server log file(s). 0=Event Not Logged and 1=Event is Logged.
Log-User-Blacklisted=0|1	If a user is blacklisted, this indicates if that information should be logged to the USR Accounting Server log file(s). 0=Event Not Logged and 1=Event is Logged.
Log-User-Password-Change=0|1	Indicates whether changes in a user's password will be logged to the USR Accounting Server log file(s). 0=Event Not Logged and 1=Event is Logged.
Log-Security-Breach=0|1	Indicates whether a security breach will be logged to the USR Accounting Server log file(s). 0=Event Not Logged and 1=Event is Logged.
[DICTIONARY]	This specifies the beginning of the section where all RADIUS dictionary files used by the server are listed. When specifying the dictionary file, include the device and directory.
n=full-path-name	This lets you specify multiple dictionary file entries, each with it's own unique number (n). At least one standard RADIUS dictionary file must be specified.
[TABLES]	This specifies the beginning of the TABLES section. This section specifies all security data files that the server uses.
table-name=full-path-name	This links a table-name with a specific data file. Table names are referenced in the authentication script file. Multiple table name entries may be specified.
[SERVER]	This specifies the beginning of the SERVER section. This section lists the primary and alternate RADIUS or Accounting servers. The U.S. Robotics' Accounting/Logging Server must be given the server-id of *Accounting*. Two entries are required for each server. The first is the server's IP address in dot notation. The second is an UDP port number, usually 1646. Multiple server entries may be specified.
server-id.IP=ipaddress	This links the IP address of the RADIUS or Accounting server with the name (server-id) used by the Security Server.
server-id.PORT=udp-port	This links the UDP port of the RADIUS or Accounting server with the name (server-id) used by the Security Server

Database and Table Files

User Table File

Each record in USERS.TXT represents a user requiring authentication. User records in the Users file contain user name, password, password expiration and possibly failed logins. The remaining lines of the record contain information that the server sends back to the NMC or NETServer client in an response packet.

Field Name	Applies To	Type	Description
Dial-In-Call-Restriction	NMC only	255 character string	This is a list of restricted/allowed dialback phone numbers issued during dial-in security sessions. The syntax for entries in this list is covered under Call Restriction List.
Dial-In-Security-Mode	NMC only	value	This identifies the dial-in security mode used for a user. Applies when the initial login sequence is passed. The default is 0 or Pass-Thru. 0 Pass-Thru: The user is connected to DTE. 1 Dialback Stored Number: The NMC disconnects and uses a stored number for dialback. 2 Dialback Entered Number: The user is prompted to supply a number for dialback.
Dial-Out-Call-Restriction	NMC only	255 character string	This is a list of restricted/allowed dial-out phone numbers. The syntax for entries in this list is shown later under Call Restriction List.
Dialback-Group-List	NMC only	255 character string	This lists dialback groups the user, or user group, is a member of. A user or user group may be a member of zero, one, or more than one user group.
Dialback-Name	NETServer only	36 character string	This is the name of a host or system in the NETServer's Location Table.
Dialback-Number	NMC and NETServer	255 character string	This field represents a standard modem dialing string used for dialing back users that are configured for Dialback-Stored Number. This can be any standard AT-type string, including plain phone numbers.
Failed-Logins	NMC and NETServer	integer	This is the number of consecutive failed login attempts made by a user since the last successful login attempt. This field is incremented by the server whenever a client request contains a valid username but an invalid password. When a user logs in successfully, this field is reset to 0.

Field Name	Applies To	Type	Description
Framed MTU	NETServer only	integer	This is the Maximum Transmission Unit allowed across the serial network connection.
Framed-Address	NETServer only	IP address	This is the IP address of the remote device.
Framed-Compression	NETServer only	value	This determines the type of compression used. Valid values are 0 (no compression) and 1 (Van Jacobson TCP/IP).
Framed-Filter-ID	NETServer only	18 character string	This indicates the packet filter that control the user's access to the network. The NETServer handles the Filter ID name internally by appending .out to identify the filter as an Output Filter, or appending .in to identify an Input Filter.
Framed-IPXNet	NETServer only	IPX network number	This is the IPX network number of the authenticated network user.
Framed-Netmask	NETServer only	IP address	This is the netmask of the remote device.
Framed-Protocol	NETServer only	value	This is the user's protocol. Valid values are 1 (PPP) and 2 (SLIP).
Framed-Routing	Response, NETServer only	value	This determines whether the serial port permits RIP packets to be sent or received across the interface. 0 None 1 Send routing packets to the network 2 Listen for routing packets from the network 3 Send and Listen
Login-Host	NETServer only	IP address	This is the IP address of the host that the user will log in to and connect with.
Login-Service	NETServer only	value	This field represents the type of connection the user will make with the remote host. Valid values are 0 through 3. 0 Telnet 1 Rlogin 2 TCP; also called Netdata 3 PortMux
Mode	NMC and NETServer	value	This defines the authentication mode for the user. Valid values are 0 (Simple Username/Password) or 1 (Security Dynamics "SecureID").
Password	NMC and NETServer	8 character string	This is the password that a user enters in response to the "password" prompt.
Password-Expire	NMC and NETServer	date	This is the password expiration date. No password history is maintained. The server updates the field when the user successfully changes a password.
Request-Dialback-Login-Validation	NMC only	value	This determines whether the security system issues a login validation sequence during dialback. Valid values are 0 or 1. The default is disabled (0) or no dialback login validation. A value of 1 means that Dialback login validation is enabled.

Field Name	Applies To	Type	Description
Request-Dialback-Modem-Selection	NMC only	value	This determines whether the NMC prompts dialback users at login for a dialback modem. Valid values are 0 or 1. The default is disabled (0). If enabled (1), the NMC prompts dialback users at login for a dialback modem. The system displays all valid choices from the list of allowed modems. Users must enter the number that appears on the menu next to their choice. If disabled, the NMC checks the Allowed Dialback modem field, polls the modems, and reserves the first available modem to dial back.
User-Group-Name	NMC only	string	This lists the user groups that a user is a member of. A user may be a member of only one or of none. A user inherits all the characteristics of the specified group. However, a group need not contain all possible user characteristics. The user may set any additional fields that the group has omitted.
User-Name	NMC and NETServer	8 character string	This contains the user's name as entered in response to the "user name" security prompt.
User-Service-Type	NETServer only	value	This field represents the type of service that this user is requesting. Note that the NETServer refers to Framed Users as Network Users, and Callback Users as Dialback Users. Valid values are 1 through 8. 1 Login 2 Framed 3 Callback Login 4 Callback Framed 5 Outbound User

Groups Table File

This file contains attribute definitions for the Groups database file. This file contains the following attributes, described in the previous section, *User Table File*:

- User-Group-Name
- Dial-In-Security-Mode
- Dial-In-Call-Restriction
- Dial-Out-Call-Restriction
- Request-Dialback-Modem-Set
- Request-Dialback-Login-Validation

- Dialback-Name-List
- Dialback-Number
- User-Service-Type
- Login-Service
- Login-Host
- Framed-Filter-Id
- Framed-Address
- Framed-Netmask
- Framed-MTU
- Framed-IPX-Network
- Framed-Protocol
- Framed-Routing
- Framed-Compression

Dialback Groups Data File

This file contains a list of all dialback modem groups.

Field Name	Type	Description
Name-Client	32 character string	This is the name of the dialback group and the client's IP address. Use the format dialback-name:client-IP-address.
Modems	17 character string	This is string of hexadecimal digits that indicate which modem channels will be used for dialback. Use the format 4nnnnnnnnnnnnnnn.

Clients Data File

The Clients data file lists the clients such as NMCs and NETServers. It contains two fields. The Security Server supports up to nine clients.

Field Name	Type	Description
IP	ip addr	This is IP address of the client. It can also be the name of the client if the client's IP address is in the TCP HOSTS file.
SECRET	16 character string	This is the RADIUS secret for the client. The first time the client requests authentication, the secret is encrypted.

Appendix E
Script Language

Script Language Syntax and Rules

Script Language Syntax

- bracketed items [ ] are considered optional.
- An ellipse (...) indicates that more than one occurrence may be used.
- <A | B> indicates a choice between A and B
- Items in **boldface** are required syntax or keywords
- Language elements that are obvious are not defined; for example, <alphanumeric string>.

General Syntax Rules

- A script file can contain comment lines starting with the pound sign (#) or a semicolon (;). All characters following the pound sign or semicolon are ignored.
- Maximum length of each line in the script file is 255 characters.
- Statements can continue onto another line by placing a backslash (\ ) at the end of the line. Make sure the backslash is not part of a commented line or the backslash will be ignored.
- Only one statement or label is permitted on a single line.
- The script language is not case-sensitive.

Script Variables and Lists

The script language permits named variables. Variable names may be any alphanumeric character including dashes (-) and underscores (_).

Maximum length is 32. Variables contain a string or numeric values. The values of string variables must be enclosed in quotes. For example:

```
ThisVariable = 20
OtherVariable = "Twenty"
```

Variable lists are set of variables, or other lists, collected together. List name and variable member are separated by a period (.) using the following format:

```
list-name.variable-name
```

For example:

```
MyList.Access.Denied.Disconnect.Reply =
   "Disconnecting Now"
MyList.Message = "Hello"
```

There is no limit on the depth of a list. Variables within a list are ordered alphabetically.

Predefined Variable Lists

DICTIONARY

This predefined list is loaded during the Security Server's initialization. It contains all attributes and associated values that are read from the dictionary(ies). For example:

```
DICTIONARY.Framed-Protocol.PPP=1
DICTIONARY.Framed-Protocol.SLIP=2
```

The exact nature of this dictionary will vary depending upon the contents of the dictionary(ies).

SYSTEM

This predefined list is created from the SYSTEM section of the Security Server Configuration File during initialization. It contains all the parameters and associated values contained in that file. For example:

```
SYSTEM.ACE$Challenge-Support=1
SYSTEM.ACE$Server-Timeout=45
```

DICTIONARIES

This predefined list contains the variables associated with each parameter specified in the DICTIONARIES section of the Security Server Configuration file.

SERVER

This predefined list contains the variables associated with each parameter specified in the SERVER section of the Security Server Configuration file. For example:

SERVER.Accounting.ip = “192.77.123.156”
SERVER.Accounting.port = 1646

TABLES

This predefined list contains the variables associated with each parameter specified in the TABLES section of the Security Server Configuration file. For example:

TABLES.USERS = “users.txt“
TABLES.CLIENTS = “clients.txt“
TABLES.DBGROUPS = “dbgroups.txt“
TABLES.GROUPS = “groups.txt“

REQUEST

This predefined list contains the variables associated with each field in the current RADIUS request packet. For example:

Variable	Description
Request.Request-Type	Numeric value of the request type.
Request.Identifier	Numeric value of the identifier.
Request.Authenticator	String of 16 characters.
Request.IP	String IP address of the requesting client.
Request.Port	Numeric port of the requesting client.

Additional variables are added to the REQUEST list based on the attributes found in the request packet. The variable names are constructed from the associated attribute contained in the dictionary file.

Referencing

Referencing Attributes

When the dictionaries are first read during the start up, a DICTIONARY variable list is created for each entry using the following format:

DICTIONARY.attribute-name[.textual-value=numeric-value]

textual-value and *numeric-value* are based on the VALUE entries in the dictionary, and may not be present for all attributes. For example, Framed-Protocol might look something like this:

DICTIONARY.Framed-Protocol.PPP=1
DICTIONARY.Framed-Protocol.SLIP=2

Referencing Records

The authorization script can read or update an field associated with a particular record match in a data file. Use the following syntax:

table-name[key-value].field-name

table-name refers to the file (Clients, Users, and so on) containing the *key-value*. The *key-value* is the record that the script searches for in the *table-name*. The script then reads or updates the *field-name*.

Referencing Server Configuration File Parameters

Configuration parameters (RADSERV.CFG) can also be referenced from the authorization script:

section-name.parameter

section-name would be, for example, SYSTEM. *parameter* could be any one of the SYSTEM parameters; for example, BlackList.

Script Execution

The authorization script is executed at startup and for each new RADIUS packet received by the server. Note, however, that script execution begins at the label associated with the event.

For example:

Event	Execution Label
At server startup time	StartUp:
At server shutdown time	ShutDown:
Receive Access-Request (1) packet	Access-Request:
Receive Access-Accept (s(packet	Access-Accept:
Receive unknown packet	UnknownPacket:

If the execution label is not found in the script, an error is reported in the log file.

Script Language Elements

Labeled Sections

Option	Description
<label>: <statement>	<label> names a section of code. It can appear on anywhere in the script, but must appear on a line by itself. An implied RETURN statement is at the end of the labeled section.

Variable References and Assignment Statements

<variable> = <expression> <list-name>.<variable>=<expression> <table-name>[<key-expression>].<field-name> = <expression>	Normal assignment statements. Variables and variable lists are declared when first assigned a value.
<expression> :=	Expressions that evaluate to a string or numeric value.
<numeric-literal>	Numeric literals are whole numbers.
| **"**<string-literal>**"**	Strings must be surrounded by double quote marks.
| <variable>	Returns the current value of the <variable> specified.
| @<variable>	Indirection operator. Returns the current value of the variable whose name is <variable>.
| <list-name>.<variable>	Return the current value of the <variable> in the <list-name>.
| <table-name>**[**<key-expression>**]**.<field-name>	Returns the value of the specified <field-name> associated with the record matching the <key-expression> in the specified <table-name>. The <table-name> must be defined in the [TABLES] section of the Server Configuration file. The <field-name> must be from the set of fields specified in the <table-name>.
| <function-call>	Return the value of the built in function.
| <expression> < **+** | **-** | ***** | **/** > <expression>	Simple integer computations. Complex multi-operator expressions are not supported. The operators must be surrounded by spaces.
	String variables can be concatenated using the + operator.

Control Statements

Option	Description
CALL <label>	A subroutine call to a label. Control is returned when a RETURN statement is executed.
RETURN	Return from a subroutine call. If RETURN is not in the subroutine, the script stops executing, and the server awaits another request to process.
BREAK	Breaks out of a FOR loop, CASE section, or WHILE loop.
EXIT	Immediately stops execution of a script. This is normally executed after a RESPOND statement. Note that this does not force the server to exit.
IF (<boolean> **)** <statement> . . . [**ELSE** <statement> . . .] **ENDIF**	If-else-end if block.
WHILE (<boolean> **)** <statement> . . . **ENDWHILE**	Executes the block of statements as long as <boolean> is TRUE. The <boolean> is evaluated at the start of the block.
<boolean> :=	All boolean expressions evaluate to TRUE or FALSE.
| <expression> [!]**InTable** <table-name>	Tests if the value of the expression can be found as a key in the specified table. Use ! for **not** in the table.
| <variable> [!]**InList** <list-name>	Tests to see if the <variable> is contained in the variable list. Use ! for **not** in the table.
| <expression> <rel-op> <expression>	Simple relational test.
<relational-operator> := { **==**, **!=**, **>**, **>=**, **<**, **<=** }	Used to compare either numeric or string expressions. ==, != equal to, not equal to >, >= greater than, greater than or equal to <, <= less than, less than or equal to
SWITCH (<expression> **)** **CASE** <literal-value>: <statement> . . . . [**CASE DEFAULT:** <statement> . . . .] **ENDSWITCH**	Switch blocks. The <expression> is evaluated at run time, and then execution is transferred to the CASE section whose <literal-value> is equal to the <expression>. If no match is found, the CASE DEFAULT section is executed, if present. A BREAK is implied at the end of each CASE section, so you cannot “fall through” to the next CASE section.
FOREACH (<variable> **IN** <list-name> **)** **ENDFOR**	Iterates across a list of variable, assigning the specific variable-name to <variable>. To reference the actual value, use the @<variable> indirection operator.

Option	Description
RESPOND (<request-type>, <string-secret-expression> [,<list-name] **)**	RESPOND sends a response packet to the client that originally sent the current request being process. <request-type> is typically passed using a reference to the DICTIONARY.Request-Type.<name>. <string-secret-expression> must be appropriate for the client receiving the response. All variables in the <list-name> with a matching entry in the RADIUS dictionary will be sent with the packet. **NOTE:** The RADIUS "Authenticator" is adjusted to reflect a response packet using the secret and MD5 encryption
SEND (<server-ip-string>, <server-port-num>, <request-type-num>, <request-id-num>, <authentication-string>, <list-name> **)**	Sends a RADIUS packet to the specified server using parameters and attributes listed in the argument. It is also used if further authentication is required by another RADIUS server. All variables in the <list-name> with a matching entry in the RADIUS dictionary will be sent with the packet. Any "Password" attribute **must** be encrypted with the MD5$() function before issuing a SEND command. **NOTE:** This function causes an Access-{type} packet being returned to the server asynchronously that will execute the *Access-{type}:* labeled section.
EXEC (<string-expression> **)**	Pass the <string-expression> to the native operation system so that it is executed. For example, EXEC("USRBEEP !!!Security Breach!!!"
EMPTY (<list-name> **)**	Empties the variable list.
REMOVE (<variable> [,<list-name>] **)**	Removes the specified <variable> from the variable from the variable list. If the variable list is not specified, the variable will be removed from the global list.
DISPLAY (<string-expression> **)**	Displays a <string-expression> in a display window on a screen or logs it to a file.
StartTimer (<id> <milliseconds> **)**	This does not interrupt any currently processing command. Starts a timer of <milliseconds> duration with <id>. When the timer fires, the section of code labeled TIMER-<id> is executed. Timers continue to fire until they are explicitly stopped. The Id must be a numeric value.
StopTimer (<id> **)**	Stops the timer with <id>.
DEGUG (SET, [No]Trace) **DEBUG (SHOW, {Variables|Compiled Script})**	Turns execution track On (SET,Trace) or Off (SET,NoTrace. Dumps the indicated internal structure to the log file or displays it on a screen.

Built-In Functions

Option	Description
ENCRYPT$ (<string-expression> **)**	Returns a string. This encrypts a string expression using an internal algorithm (not MD5).
DECRYPT$ (<string-expression> **)**	Returns a string. This decrypts a string expression using an internal algorithm (not MD5).
MD5$ (<string-authenticator-expression>, <string-secret-expression>, <string-password-expression> **)**	Returns a string associated with applying the MD5 encryption algorithm to the password string. If the string is already encrypted, this variable returns clear-text; otherwise it encrypts the password.
MD5CHAP (<string-challenge>, <string-Chap-Password-expression>, <string-password-expression> **)**	Returns a numeric. This returns a 1 if the CHAP response matches the CHAP challenge, and returns a 0 otherwise. The Chap-Password typically contains the 1-character Chap-Id followed by the 16-octet response. The 16 octet response is typically the Request.Authenticator attribute. The password must be clear text.
LENGTH (<string-expression> **)**	Returns the number of characters in a string-expression.
FORMAT$ (<format-string-expression> [,<variable> . . .] **)**	Returns a string based on the <format-string-expression> and the values of all variables listed. The <format-string-expression> is identical to that used in the C sprintf function. String substitution **must** use %s, while all numeric values **must** use %ld (long signed integers).
NUM (<string-expression> **)**	Returns a numeric value of the string expression.
STR$ (<numeric-expression>)	Returns a string value of the numeric expression.
INSTR (<string>, <search-string> **)**	Searches <string> for <search-string> and returns a 0 if the <search-string> is not found. Otherwise it returns the position of the first letter of the <string> in the <search-string>.
MID$ (<string>, <start> [,<size>] **)**	Returns a substring of <string> starting in the position specified by <start>. The size of the substring is determined by <size>.
DATE (<date-string> **)**	Returns the number of seconds since midnight January 1, 1970. The <date-string> must be in format mm/dd/yyyy [hh[:mm[:ss]]].
DATE$ (<date-number> **)**	Returns the date in the form of mm/dd/yyyy hh:mm:ss based on seconds since midnight January 1, 1970.
NOW ( )	Returns the number of seconds since midnight January 1, 1970.
ASK$ (<prompt-string> **)**	This function interactively prompts with the <prompt-string>, and returns the value entered as a string expression. In Windows, a dialog box is used. In UNIX, stdin is used.

Security Dynamics Functions

Option	Description
SD$CHECK (<user-name-string>, <passcode-string> **)**	Request that the <user-name-string> and <passcode-string> be checked for authorization. After a call to SD$CHECK, the following new variables are available to the script: ACE$SD.MIN_PIN_LEN: The minimum number of characters in a PIN. ACE$SD.MAX_PIN_LEN: The maximum number of characters in a PIN. ACE$SD.USER_SELECTABLE: If set to 0, the ACE/Server generates the PIN. If set to 1, the user may set the PIN. ACE$SD.ALPHANUMERIC: If set to 0, the PIN must be numeric. If set to 1, an alphanumeric PIN is acceptable. ACE$SD.SYSTEM_PIN: The ACE/Server generated PIN if ACE$SD.SELECTABLE is set to 0. When the authorization check is finished, SD$CHECK returns one of the following numeric values: ACM-OK ACM_ACCESS_DENIED ACME_NEXT_CODE_REQUIRED ACM_NEW_PIN_REQUIRED If the messages ACM_NO_CONFIG_FILE or ACM_NO_UDP_PORT are returned, the system does not have access to an ACE/Server.
SD$NEXT (<user-name-string>, <passcode-string> **)**	This function is issued in response to a ACM_NEXT_CODE_REQUIRED message. It is used to check the next passcode/password entered by the user. Returns either ACM_NEXT_CODE_OK (user is authorized) or ACM_NEXT_CODE_BAD (no authorization). The messages ACM_NO_SESSION is returned if there was no previous call to SD$CHECK with the same user name.
SD$PIN (<user-name-string>, <pin-string> **)**	This function is issued in response to a ACM_NEXT_PIN_REQUIRED message to reset the PIN. Returns either ACM_NEW_PIN_ACCEPTED or ACM_NEW_PIN_REJECTED. The messages ACM_NO_SESSION is returned if there was no previous call to SD$CHECK with the same user name.

Index

N

O

P

R

S

T

U

X