

IBM eNetwork Firewall for AIX



Reference

Version 3 Release 2.2

IBM eNetwork Firewall for AIX



Reference

Version 3 Release 2.2

Note

Before using this information and the product it supports, be sure to read the general information under "Notices" on page 195 .

Fourth Edition (August 1998)

This edition applies to Version 3 Release 2.2 of the IBM eNetwork Firewall for AIX (product number 5765-C16). This edition replaces SC31-8418-01.

Portions Copyright © 1993, 1994 by NEC Systems Laboratory.

Contains security software from RSA Data Security, Inc. Copyright © 1990, 1995 RSA Data Security, Inc. All rights reserved.

© **Copyright International Business Machines Corporation 1994, 1998. All rights reserved.**

Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Contents

About This Book	vii
Prerequisite Knowledge	vii
Enhancements	vii
Simple Administration	vii
Network Security Auditor	vii
AIX 4.2.1 and 4.3 Support	viii
National Language Support	viii
IBM Firewall Installable Units	viii
Entering IP Addresses	ix
How to Call IBM for Service.	ix
Chapter 1. Using the IBM Firewall Command Line Interface.	1
Adapters	1
Configuration Server	2
Connections	2
Domain Name Services	5
Enterprise Firewall Manager.	6
File System Integrity Checker	8
Filters	8
Filter Rules	9
Filter Rule Checker	11
HTTP Proxy	11
Log Archiver	12
Log File Management	12
Log Monitor.	14
Mail	16
Network Address Translation	17
Network Objects	19
Network Object Groups	21
Paging	23
Pager Configuration.	23
Carrier	24
Modem Configuration	25
Testing Pager Configuration	27
Multiple Pagers	27
Security Policies	28
Services	28
Socks Rules	31
Tunnels	32
Users	35
Chapter 2. Using Report Utilities	43
Report Utilities Usage	44
IBM Firewall Log Format	44
Generating Messages from the Firewall Log File	44
Extracting Firewall Log Files	45
Generating Database Import Files	45
Using a Database with Report Utilities	47
User Interface into Report Utilities	48
The SQL Tables	52
Chapter 3. Providing Your Own Authentication Methods	65
User-Supplied Authentication	65

FWUSERPT and FWUSERAU Specifications	65
Example of fwuserpt and fwuserau	66
Secure Key as an Example of User-Supplied Authentication	66
User-Supplied Iteration Support	66
Library Requirements	67
Details of fw_prompt	68
Details of fw_tn_authenticate	68
Details of fw_ftp_authenticate	70
Sample Makefile for fwuser.o	72
Chapter 4. Using Network Management with the IBM Firewall	75
SNMP - Management Information Base (MIB)	77
Chapter 5. Using the Make Key File Utility (MKKF)	79
Creating a key file	79
Chapter 6. Troubleshooting and Testing	87
Installation and Setup	87
Filter support fails	87
Routing Problems	87
Cannot ping hosts from the firewall	87
Cannot ping nonsecure hosts from secure hosts (or vice-versa)	88
DNS Problems	89
DNS has not been configured yet.	89
File permissions do not allow the specified action.	89
nslookup www.ibm.com. nns.nns.nns.nns fails	90
nslookup www.ibm.com. 127.0.0.1 fails.	90
nslookup host.secure.company.com. sns.sns.sns.sns fails.	90
nslookup www.ibm.com. sns.sns.sns.sns fails	90
Configuration Client	90
Server not responding	90
Unable to log on to the Configuration Server	91
Traffic Control	92
Filters are damaged and contact to the manager is lost.	92
Changes to fwfilters.cfg are lost	92
Changes made to Connections do not take effect.	92
Proxy Servers	93
No data transmitted.	93
Cannot connect to the desired host	93
Idle telnet proxy users are not being timed out	93
Network Address Translation	94
Ping does not work on my NAT connection	94
The NAT connection does not work	94
How can a route be established for NAT packets?	94
How do I add an arp entry?	94
What is the syntax for the arp command?.	94
How should I add the arp command?	95
What debugging tools are available to help with NAT?	95
How do I run IPtrace?	95
Win 95 Remote IPsec Client	97
Unexpected characters received when using telnet	97
Log Facilities	97
Log facility changes do not take effect on the server	97
Report Utilities.	97
Failure on system call: ar -vt	97
An error occurred while accessing the file:	97

Errors occur importing data to the database.	97
Log Management	98
fwlogmgmt -l fails	98
Log archival never happens.	98
Testing the Firewall Ports Using fwice	98
The hosts file	99
The services file	99
The results file.	99
Appendix A. Messages	101
Message Tag	101
Messages	101
Appendix B. Hardening for the AIX System Configuration	153
Appendix C. SNMP Management Information Base (MIB).	155
Appendix D. ESP Specification for CDMF	181
Keys	181
Initialization Vector	181
Data Size	181
Payload Format	181
Algorithm.	183
Encryption	183
Decryption	183
Security Considerations	184
Appendix E. Obtaining Requests for Comments (RFCs)	185
Appendix F. Creating a Socks Configuration File	187
Example of a sockd Rule.	189
Appendix G. The Crontab Command.	191
Bibliography	193
Information in IBM Publications	193
Firewall Topics.	193
Internet and World Wide Web Topics	193
General Security Topics	193
Information in Industry Publications	193
Notices	195
Trademarks.	195
Glossary	197
Index	199
Readers' Comments — We'd Like to Hear from You.	201

About This Book

This book is intended as a reference for network or system security administrators who install, administer, and use the IBM eNetwork Firewall Version 3.2.2 on an AIX/6000. To use client programs such as Telnet or FTP, please see the user's guide for your TCP/IP client programs.

Prerequisite Knowledge

It is important that you have a sound knowledge of TCP/IP and network administration before you install and configure the IBM eNetwork Firewall. Because you will set up and configure a firewall that controls the access in and out of your network, you must first understand how the network operates. Especially, you need to understand the basics of IP addresses, fully qualified names, and subnet masks.

An excellent book on TCP/IP that covers netstat, arp, ifconfig, ping, nslookup, DNS, sendmail, routing, and much more is *TCP/IP Network Administration*. See the *Bibliography* for more details.

An excellent book for those performing UNIX administration, that also gives an excellent overview of TCP/IP and routing, network hardware, DNS, and sendmail is the *UNIX System Administration Handbook*. See the *Bibliography* for more details.

Enhancements

The IBM eNetwork Firewall for AIX offers several enhancements.

Simple Administration

Through use of a Java** application, which you can administer from a remote machine, you can easily make updates to the firewall configuration. And, different administrators can be assigned different levels of authority to further control access to the firewall. This single, easy-to-understand graphical user interface (GUI) can be used to administer both the AIX Firewall and the Windows NT Firewall.

Network Security Auditor

The Network Security Auditor (NSA), a tool that checks your network for security holes or configuration errors, has been enhanced. NSA:

- Is faster and more robust
- Supports delta report generation
- Offers enhanced site policy facility
- Supports additional servers
- Has additional security vulnerability checks

By periodically running the Network Security Auditor, you can ensure that nothing has been modified in a way that creates a security vulnerability especially after you put the Firewall on-line.

AIX 4.2.1 and 4.3 Support

AIX 4.2.1 and 4.3 are supported, exclusive of the AIX Common Desktop Environment. Previous releases of AIX are no longer supported.

National Language Support

National language support is offered for English, Japanese, Korean, French, simplified Chinese, traditional Chinese, Italian, Spanish, and Brazilian Portuguese.

IBM Firewall Installable Units

The IBM Firewall separate installable components are:

- EFM
 - IBM Enterprise Management System (a firewall that manages other firewalls)
- FW
 - Base IBM Firewall
 - IBM Firewall Common Libraries and Catalogs
 - IBM Firewall Remote Configuration Client
 - IBM Firewall Report Generation Utilities
- Netscape.NAV
 - Netscape Navigator**
- ipsec
 - IPSec Client
- nsauditor
 - Network Security Auditor
 - Network Security Auditor HTML Interface
- sva
 - System View Agent for AIX
 - SystemView Agent for AIX SNMP Mapper
- sway
 - General export and domestic customization files
 - IBM KeyWorks
 - Key Recovery Service Provider

For directions on how to install the Windows 95 secure remote client, refer see the *IBM eNetwork Firewall User's Guide*.

To install the PDF version of this manual and the *IBM eNetwork Firewall User's Guide* download the following files from the fwbooks directory on the IBM Firewall CDROM to your workstation:

- fwuser.pdf
- fwref.pdf

Use the Adobe Acrobat** Reader to view these books. If you do not have the Adobe Acrobat Reader installed, you can go to the Adobe Web site at:

<http://www.adobe.com/prodindex/acrobat/> to learn more about the Adobe Acrobat Reader and to get a copy.

Entering IP Addresses

When you configure your firewall, you will be asked to enter IP addresses. You should enter a complete dotted-decimal IP address, with all 4 octets, in the format:

nnn.nnn.nnn.nnn

where each nnn is a set of three numbers in the range 000–255.

How to Call IBM for Service

The IBM Support Center provides you with telephone assistance in problem diagnosis and resolution. You can call the IBM Support Center at any time; you will receive a return call within eight business hours (Monday–Friday, 8:00 a.m.–5:00 p.m., local customer time). The number to call is 1-800-237-5511.

Outside the United States or Puerto Rico, contact your local IBM representative or your authorized IBM supplier.

Chapter 1. Using the IBM Firewall Command Line Interface

If you have purchased the No Encryption version of the IBM eNetwork Firewall for AIX, the tunnel commands in this chapter do not apply.

This chapter discusses commands that you can use from an IBM eNetwork Firewall command line.

The following information applies to the commands:

- The commands listed in this book use the following syntax:
 - underlined indicates this is user-entered data.
 - [] indicates a parameter is optional.
 - {} indicates the user has a choice of parameters.
 - | separates choices.
- All parameters use a keyword=value format.
- If a parameter has multiple values the values should be within double quotes and be delimited by blank spaces, for example:
secaddr="11.22.33.1 11.22.33.2"
- Do not include spaces inside any parameter unless it is within double quotes.
- If you omit one or more required parameters, the command-line utility lists missing parameters.
- If an invalid value for a parameter is entered, the command-line utility reports this error.
- Some of the firewall daemons dynamically update their behavior when their configuration files change. Some require an update subcommand. An update subcommand is provided for those daemons that require an instruction.
- Only firewall administrators can execute programs from the command line.
- Because of the complexity and file interdependencies, **do not directly edit any configuration files.**

Adapters

To list the adapters, issue the following command.

```
fwadapter cmd=list  
[addr=AdapterAddress]
```

addr=AdapterAddress

Lists all adapters attached to this machine and identifies each as being either a secure or a nonsecure adapter. If the optional addr parameter is specified, only that adapter is listed.

To change the adapters, issue the following command.

```
fwadapter cmd=change  
addr=AdapterAddress  
state={secure|nonsecure}
```

The parameter definitions are:

addr=AdapterAddress

Address of the adapter to change.

state={secure|nonsecure}

Sets the secure/nonsecure state of the adapter answering to the specified IP address.

Configuration Server

The `fwcfgsrv` command lists or changes the configuration server's options. An administrator must have the authority to administer traffic control functions to issue this command.

To list the configuration server options, issue the following command.

```
fwcfgsrv cmd=list
```

The output from the `fwcfgsrv` command looks as follows:

```
localonly = yes/no
encryption = none/ssl
sslfile = filename if one is defined
```

To change the configuration server options, issue the following command.

```
fwcfgsrv cmd=change
    [localonly={yes|no}]
    [encryption={none|ssl}]
    [sslfile=]
```

The parameter definitions are:

localonly

Indicates if the firewall can only be administered from a local machine. Valid values are yes or no.

encryption

Indicates if the configuration server expects incoming data to be encrypted through ssl or not. Valid values are none or ssl.

sslfile Indicates the ssl key file name to be used for ssl encryption. See "Chapter 5. Using the Make Key File Utility (MKKF)" on page 79.

Connections

The `fwconns` command is used to create, modify, delete and list firewall connections. These connections associate network objects with services and/or socks templates to define the types of communications allowed between endpoints. For more information on connections, see the *IBM eNetwork Firewall User's Guide*.

To create firewall connections, issue the following command.

```
fwconns cmd=create
    name=_name_
    [desc=_description_]
    source=network_id_or_name
    [destination=network_id_or_name]
    {servicelist=service_id1,service_id2,...,service_idn |
     sockslist=socks_id1,socks_id2,...,socks_idn}
```

The parameter definitions are:

name Is the name you want to give to this connection.

desc Is a description of this connection.

source

Is the source network object name or ID number for the connection.

destination

Is the destination network object name or ID number for the connection.

servicelist

Is a list of services by ID number to include in this connection.

sockslist

Is a list of socks rules by ID number to include in this connection.

To modify firewall connections, issue the following command.

```
fwconns cmd=change
  {name= name_ | id= id_}
  [desc=[_description_]]
  [source=network_id_or_name]
  [destination=[network_id_or_name]]
  [servicelist=[service_id1,service_id2,...,service_idn]]
  [sockslist=[socks_id1,socks_id2,...,socks_idn]]
```

The parameter definitions are:

name Is the name of the connection you want to change.

id Is the ID number of the connection you want to change.

desc Is the new description of this connection. If this description parameter is supplied with no value, the description is removed.

servicelist

Is a list of services by ID number to include in this connection.

sockslist

Is a list of socks rules by ID number to include in this connection.

To delete firewall connections, issue the following command.

```
fwconns cmd=delete
  {name=_name_ | id=_id_}
```

The parameter definitions are:

name Is the name of the connection you want to delete.

id Is the ID number of the connection you want to delete.

To list firewall connections, issue the following command.

```
fwconns cmd=list
  [name=_name_ | id=_id_]
  [format={short|long|wide|raw}]
```

The parameter definitions are:

name Is the name of the connection you want to list.

id Is the ID number of the connection you want to list. If you do not supply a name or an id, all connections are listed.

format

Describes the format of this command's output.

To move firewall connections, issue the following command.

```
fwconns cmd=move
        connection=connection_id_or_name
        after=[connection_id_or_name]
```

The parameter definitions are:

connection

Is the name or ID number of the connection you want to move.

after Is the name or ID number that the moved connection should follow.

To add a firewall service to a connection, issue the following command.

```
fwconns cmd=addservice
        servicelist=service_id1,service_id2,...,service_idn
        connection=connection_id_or_name
```

The parameter definitions are:

servicelist

Is a list of services by ID number that you want to add to the connection.

connection

Is the name or ID number of the connection you want to change.

To remove a firewall service from a connection, issue the following command.

```
fwconns cmd=removeservice
        servicelist=service_id1,service_id2,...,service_idn
        connection=connection_id_or_name
```

The parameter definitions are:

servicelist

Is a list of services by ID number that you want to remove from the connection.

connection

Is the name or ID number of the connection you want to change.

To move a firewall service within a connection, issue the following command.

```
fwconns cmd=moveservice
        service=service_id_or_name
        after=[service_id_or_name]
        connection=connection_id_or_name
```

The parameter definitions are:

service

Is the name or ID number of a specific service that you want to move within the connection.

after Is the name or ID number that the moved service should follow.

connection

Is the name or ID number of the connection you want to change.

To add socks rules within a connection, issue the following command.

```
fwconns cmd=addsocks
        sockslist=socks_id1,socks_id2,...,socks_idn
        connection=connection_id_or_name
```

The parameter definitions are:

sockslist

Is a list of socks rules by ID number that you want to add to the connection.

connection

Is the name or ID number of the connection you want to change.

To remove socks rules within a connection, issue the following command.

```
fwconns cmd=removesocks
sockslist=socks_id1,socks_id2,...,socks_idn
connection=connection_id_or_name
```

The parameter definitions are:

sockslist

Is a list of socks rules by ID number that you want to remove from the connection.

connection

Is the name or ID number of the connection you want to change.

To move socks rules within a connection, issue the following command.

```
fwconns cmd=movesocks
socks=socks_id_or_name
after=[socks_id_or_name]
connection=connection_id_or_name
```

The parameter definitions are:

socks Is the name or ID number of a socks rule you want to move within the connection.

after Is the name or ID number of a socks rule that the moved socks rule should follow.

connection

Is the name or ID number of the connection you want to change.

Domain Name Services

The Domain Name Service (DNS) provides full domain name service to hosts inside the secure network while providing minimal information to hosts outside the secure network. Three domain name servers are required to accomplish this:

- One at the firewall
- One inside the secure network
- One outside the secure network.

See the *IBM eNetwork Firewall User's Guide* for more information.

Note:

1. The x.x.x.x is an IP address in its dotted decimal format.
2. The value for the secaddr and remaddr parameters can be a single IP address or a list of IP addresses. If a list of IP addresses is specified, the list should be space delimited and contained within double quotes.
3. Duplicate addresses are detected and flagged as an error.
4. The first time DNS is configured, fwdns cmd=change creates the new file. The firewall will always have exactly one DNS configuration record. The

values may be empty. The change subcommand is sufficient to change any or all of the values in the DNS record.

The following command lists the current DNS configuration.

```
fwdns cmd=list
```

To change the DNS configuration entry and create a new file:

```
fwdns cmd=change
  secdomain=SecureDomainName
  secaddr=x.x.x.x | "x.x.x.x x.x.x.x x.x.x.x"
  remaddr=x.x.x.x | "x.x.x.x x.x.x.x x.x.x.x"
```

The parameter definitions are:

secdomain=*SecureDomainName*
domain name of your internal, secure network

secaddr=*SecureDNSAddr[,...]*
IP address of your secure domain name servers

remaddr=*NonSecureDNSAddr[,...]*
IP address the domain name servers outside your secured network that are provided by your Internet connection service provider.

Enterprise Firewall Manager

Enterprise Firewall Manager (EFM) allows for selective management of one firewall's configuration file from another firewall.

To read the EFM's security agreement record for the specified firewall and get a list of functions that can be administered at the EFM, use the following command.

```
fwmanager cmd=list
  type=secagree
  firewallname=FirewallName
```

The parameter definitions are:

cmd=list
Lists each function in the security agreement and indicates whether the EFM or the managed firewall can configure that function.

type=secagree
the security agreement.

firewallname=*FirewallName*
The name of the firewall.

To return a list of functions that have been modified since the last time configuration files were downloaded for the firewall use the fwtransfer command. The administrator can also use this command to confirm changes prior to initiating the fwtransfer cmd=transfer type=changed command.

```
fwtransfer cmd=list
  type=changed
  firewallname=FirewallName
```

The parameter definitions are:

cmd=list

Returns a list of functions that have been modified since the last time configuration files were downloaded for the Firewall.

type=changed

Lists the functions that have had configuration file changes.

firewallname=Firewallname

The name of the firewall.

To download configuration files for requested functions to the specified firewall use the following command.

```
fwtransfer cmd=transfer
          type=changed
          firewallname=FirewallName
          service=[all |"DNS sendmail networkobj traffic NAT VPN
          pagersup interface logmonitor proxyadmin logfacility secagree SNMP"]
```

The parameter definitions are:

type=changed

Configuration files that have been modified for requested services if a change has been made to the file since the last time the file was downloaded.

firewallname=Firewallname

The name of the firewall.

service=

Specifies services. The default is all.

The secagree service type also includes session limit changes.

To transfer configuration files regardless of whether they have changed, use the following command.

```
fwtransfer cmd=transfer
          type=select
          firewallname=FirewallName
          service=[all |"DNS sendmail networkobj traffic NAT
          VPN pagersup interface logmonitor proxyadmin
          logfacility secagree SNMP"]
```

The parameter definitions are:

type=select

Forces the download of configuration files for specified services.

firewallname=Firewallname

The name of the firewall.

service=

Specifies services. The default is all. The secagree service type also includes session limit changes.

To activate configuration files previously downloaded to the managed firewall, use the following command.

```
fwact firewallname=FirewallName
          service=[all |"DNS sendmail networkobj traffic
          NAT VPN pagersup interface logmonitor
          proxyadmin logfacility secagree SNMP"]
```

The parameter definitions are:

firewallname=*Firewallname*

The name of the firewall.

service=

Specifies services. The default is all. The secagree service type also includes session limit changes.

File System Integrity Checker

The following command invokes the File System Integrity Checker.

```
fwfschk  
[cmd={-? | -l |  
-u | -f}]
```

The parameter definitions are:

fwfschk -?

usage

fwfschk -l

logs output

fwfschk -u

updates database

fwfschk -f

forces the update to the database

Filters

Use the `fwfilter` command to activate and deactivate filter rules.

```
fwfilter cmd=update | verify | list | shutdown | startlog |  
stoplog
```

The parameter definitions are:

fwfilter cmd=update

rebuilds the configuration and activates that rule set.

fwfilter cmd=verify

performs a "test build" of the configuration but does not activate any changes.

fwfilter cmd=list

lists the most recently built configuration

fwfilter cmd=shutdown

deactivates the filters mechanism

fwfilter cmd=startlog

logs selected traffic to the firewall log facility

fwfilter cmd=stoplog

stops the firewall filter logging

Filter Rules

Use the `fwfrule` command to create, modify, delete, and list rule templates used to create a service. See the *IBM eNetwork Firewall User's Guide* for a full explanation of the parameters used on this command

To add rule templates, use the following command.

```
fwfrule cmd=add
  name=name
  [desc=description]
  type={permit|deny}
  protocol={all|tcp|ack|udp|icmp|ospf|ipip|esp|ah}
  [srcopcode={any|eq|neq|lt|gt|le|ge}]
  [srcport=port_num]
  [destopcode={any|eq|neq|lt|gt|le|ge}]
  [destport=port_num]
  interface={both|secure|nonsecure}
  routing={both|local|route}
  direction={both|inbound|outbound}
  [log={yes|no}]
  [tunnel=tunnel_id]
  [fragment={yes|no|only|headers}]
```

The parameter definitions are:

name Is the name you want to give to this filter rule.

desc Is the description of the filter rule.

type Indicates if the rule will permit or deny traffic.

protocol
Is the protocol type for this rule.

srcopcode
Indicates the logical operation to be performed on the source port number.

srcport
Indicates the source port number for this rule (0–65535).

destopcode
Indicates the logical operation to be performed on the destination port number.

destport
Indicates the destination port number for this rule (0–65535).

interface
Indicates which interfaces this rule applies to.

routing
Indicates the destination of traffic for this rule.

direction
Indicates the direction of traffic for this rule.

log Indicates whether or not log records will be written when this rule is matched.

tunnel Indicates a tunnel id for this rule (1–999999).

fragment
Indicates the type of fragment control for this rule.

To modify rule templates, use the following command.

```
fwfrule cmd=change
  name=name | id=ruleid
  [desc=[description]]
  [type={permit|deny}]
  [protocol={all|tcp|ack|udp|icmp|ospf|ipip|esp|ah}]
  [srcopcode=[{any|eq|neq|lt|gt|le|ge}]]
  [srcport=[port_num]]
  [destopcode=[{any|eq|neq|lt|gt|le|ge}]]
  [destport=[port_num]]
  [interface={both|secure|nonsecure}]
  [routing={both|local|route}]
  [direction={both|inbound|outbound}]
  [log=[{yes|no}]]
  [tunnel=[tunnel_id]]
  [fragment=[{yes|no|only|headers}]]
```

The parameter definitions are:

- name** Is the name of the filter rule you want to change.
- id** Is the ID number of the filter rule you want to change.
- desc** Is the description of the filter rule.
- type** Indicates if the rule will permit or deny traffic.
- protocol**
Is the protocol type for this rule.
- srcopcode**
Indicates which source port numbers apply to this rule.
- srcport**
Indicates the source port number for this rule (0–65535).
- destopcode**
Indicates which destination port numbers apply to this rule.
- destport**
Indicates the destination port number for this rule (0–65535).
- interface**
Indicates which interfaces this rule applies to.
- routing**
Indicates the destination of traffic for this rule.
- direction**
Indicates the direction of traffic for this rule.
- log** Indicates whether or not log records will be written when this rule is matched.
- tunnel** Indicates a tunnel id for this rule (1–999999).
- fragment**
Indicates the type of fragment control for this rule.

To delete rule templates, use the following command.

```
fwfrule cmd=delete
  name=name | id=ruleid
  [force={yes|no}]
```

The parameter definitions are:

- name** Is the name of the filter rule you want to delete.
- id** Is the ID number of the filter rule you want to delete.
- force** Indicates whether or not this rule should be deleted even if other objects are dependent upon it.

To list rule templates, use the following command.

```
fwfrule cmd=list
      [name=name] | [id=ruleid]
      [format={short|long|wide|raw}]
```

The parameter definitions are:

- name** Is the name of the filter rule you want to list.
- id** Is the ID number of the filter rule you want to list.
- format**
Describes the format of the command's output.

Filter Rule Checker

The following command is used to test the firewall's filter rules.

```
fwice [hosts=HostsFileName]
      [services=ServicesFileName]
      [results=ResultsFileName]
```

See "Testing the Firewall Ports Using fwice" on page 98 for more information. This utility is run from a crontab. See "Appendix G. The Crontab Command" on page 191 for more information.

HTTP Proxy

HTTP proxy efficiently handles browser requests through the IBM Firewall eliminating the need for a socks server for Web browsing. Users can access useful information on the Internet, without compromising the security of their internal networks and without altering their client environment to implement HTTP proxy.

The fwhttp command lists or changes the current HTTP proxy configuration.

To list the current HTTP proxy configuration, use the following command.

```
fwhttp cmd=list
```

To change the current HTTP proxy configuration, use the following command.

```
fwhttp cmd=change
      [port=]
      [maxcontentlengthbuffer=]
      [minactivethreads=]
      [maxactivethreads=]
      [idlethreadtimeout=]
      [logging=]
```

The parameter definitions are:

- port** The port on which the http proxy service will listen.

maxcontentlengthbuffer

The maximum size of a buffer for returning documents to allow the addition of a content-length header to be returned.

minactivethreads

The minimum number of worker threads to start at initialization and to keep alive at run time.

maxactivethreads

The maximum number of worker threads that can be running at any time.

idlethreadtimeout

The length of time to keep idle threads available.

logging

Indicates if logging is desired for HTTP activity. Values are on or off.

Log Archiver

The following command invokes the logfile archiver to maintain log facilities that have been configured for archiving.

```
fwlogmgmt -l or fwlogmgmt -a
```

It is useful to put this command in a scheduled cron job. See the *IBM eNetwork Firewall User's Guide* for more information.

Log File Management

Log file management defines and manages your log and archive files. The fwlog command adds, modifies and deletes log facilities.

To add log facilities, issue the following command.

```
fwlog cmd=add
    facility=Facility
    priority=Priority
    logfile=LogFile
    [arcfile=ArchivePath
    logtime=DaysToKeepInLog
    arctime=DaysToKeepInArchive
    workspace=workspace directory]
```

Valid values for **facility**:

- firewall (local4) - general firewall logs including filter logging
- alert (local1) - log monitor daemon status and threshold violation warnings used to populate the Alerts Display
- mail - mail logs
- syslog - useful in case the other logs fill up their file systems
- * - all facilities

Valid values for **priority**:

- debug
- info
- notice
- warning
- err

- crit
- alert
- emerg

The logfile parameter indicates where the syslog entries should be sent. Valid values for logfile are:

- A fully qualified file name (starting with a '/' character), indicating the file to which the log entries should be written
- A host name prefixed with an '@' sign
- A user name on the firewall or a comma-delimited list of user names on the firewall.

Note: Files identified for the alert log or the firewall log facilities should be different from each other and different from the files for any other log facility if firewall features will be used to process these files.

It is important that ONLY firewall log messages appear in files input to report utilities. No other facility should be directed to the same file as the firewall log or alert log.

These parameters control log archival. For actual log archival to occur, the fwlogmgmt command must be run periodically. See the *IBM eNetwork Firewall User's Guide* for more information.

The arcfile, logtime, arctime, and workspace parameters are optional, and are only valid when the logfile parameter specifies a file name. All four parameters must be specified if any are specified. These parameters control log archival. For actual archival to occur, run the fwlogmgmt command periodically. See "Log Archiver" on page 12 .

The **arcfile** parameter must contain a fully qualified path and file name.

The **logtime** parameter indicates the minimum number of days a syslog entry will remain in the logfile before being moved to the archive file.

The **arctime** parameter indicates the minimum number of days a syslog record will remain in the archive file before being purged.

The **workspace** parameter specifies a directory the archiving program should use for temporary work files when archiving syslog files.

To change log facilities, issue the following command.

```
fwlog cmd=change
  index=Index
  [facility=Facility]
  [priority=Priority]
  [logfile=LogFileName]
  [arcfile=ArchiveFileName]
  [logtime=DaysToKeepInLog]
  [arctime=DaysToKeepInArchive]
  [workspace=WorkspaceDirectory]
```

If a change, particularly the initial instance, fails to create a syntactically correct configuration file (for example, the log file definition that was created has missing fields), a warning is issued and the firewall will not log data.

To perform logging but no archiving, only the **facility**, **priority**, and **logfile** parameters are required. To disable log archival once it is started, blank out the **archive**, **logtime**, and **arctime** parameters. If you have scheduled an archival job, delete it.

To list the current log-file configuration data, issue the following command.

```
fwlog cmd=list
```

To delete the firewall log entry specified by the index number returned for the entry on the fwlog cmd=list command, issue the following command.

```
fwlog cmd=delete
      index=index of entry to delete
```

Log Monitor

Use the log monitor command to tell the log monitor when and how to trigger alerts. Alerts occur when threshold values specified in this command (or the corresponding configuration client panel) are reached within a specified time interval. When an alert occurs:

1. A record is written to the firewall alerts facility and to the firewall logging facility
2. A specified command is run
3. A notice is sent to one or more user IDs
4. A message is sent to a paging device

The last three actions are controlled by proper configuration of values specified here.

Listing the Log Monitor Settings

```
fwlogmon cmd=list
```

Specifying User IDs to Receive Mail Notifications when any Alert Occurs

To specify user ids to receive mail notifications when any alert occurs (the notice is sent to each id you add):

```
fwlogmon cmd=add|delete
          type=id
          username=
          [comment=]
```

Specifying a Command to be Run when any Alert Occurs

```
fwlogmon cmd=add|change
          type=command
          command=
          [comment=]

fwlogmon cmd=delete
          type=command
```

Specifying a Threshold at which an Alert Should be Triggered Based on the Number of Unsuccessful Login Attempts

```

fwlogmon cmd=add
         type=single|multi|host
         count=
         time=
         pager=
         [comment=]

fwlogmon cmd=change
         type=single|multi|host
         [count=]
         [time=]
         [pager=]
         [comment=]

fwlogmon cmd=delete
         type=single|multi|host

```

Specifying a Threshold at which an Alert Should be Triggered Based on Number of Occurrences of a Specific Firewall Message ID

```

fwlogmon cmd=add
         type=msg
         tag=
         count=
         time=
         pager=
         [comment=]

fwlogmon cmd=change
         type=msg
         tag=
         [count=]
         [time=]
         [pager=]
         [comment=]

fwlogmon cmd=delete
         type=msg
         tag=

```

The parameter definitions are:

- type** Identifies the type of log monitor command characteristic being added or modified.
Allowed values are id, command, msg, single, multi, and host.
- id** Affects the user id to send notices to.
- command** Specifies a command to be executed.
- msg** Affects the monitoring of a specific log message.
- single** Affects monitoring based on single user ids. A counter is kept for each id that has a failed attempt. If the counter for any id reaches the threshold value specified in this command, an alert is triggered.
- multi** Affects monitoring based on multiple user ids. If the total of all the counters, for all user ids that have had failed attempts, reaches the threshold value specified in this command, an alert is triggered.
- host** Affects monitoring based on host names. A counter is kept for each host name from which a failed attempt occurs. If the counter for any host name reaches the threshold value specified in this command, an alert is triggered.

username

The mail id of a firewall administrator or other user to be notified of any alert. Alert notifications will be successfully mailed only if you have properly configured a secure-side mail server.

command

The name of the command to be executed when any alert occurs. It must be the full-path name of an executable file. It can be a .bat file, allowing multiple commands to be executed from within that file, however if the .bat file makes any reference to other files, they also must be full-path name references.

count Sets the threshold for the number of failures, or occurrences of a particular log message, at which an alert will be used.

time Sets a time-interval in minutes. The count must be reached within this interval of time from the first occurrence, in order for an event to be triggered. Occurrences older than this interval before the current time are dropped from the count.

pager Specifies whether you use a page or not, when the associated threshold triggers an alert. The active pager configuration is used to send the page.

tag A log message tag (with the message prefix ICA) to be monitored. Log monitor messages (ICA tags lower than 1000) cannot be monitored.

Mail

Use the `fwmail` command to map public and secure mail domains.

```
fwmail cmd=list
fwmail cmd=add
    secdomain=
    mail=
    remdomain=
fwmail cmd=change
    secdomain=
    [mail=]
    [remdomain=]
fwmail cmd=delete
    secdomain=
```

The parameter definitions are:

secdomain

The name by which the mail domain being described is known to users on the secure side of the firewall.

mail Address of a mail server.

remdomain

The name by which the mail domain being described is known to users on the nonsecure side of the firewall.

Network Address Translation

Network address translation (NAT) provides a solution to the IP address depletion problem by allowing addresses inside your secured IP network to be reused by any other IP network.

NAT supports four types of configuration:

- **Many-to-One Registered Address** - Many-to-one translation involves translating a packet's secure address and port number such that many (up to 65536) internal addresses can share one registered IP address. This one shared registered IP address will hide local addresses but in addition to it, you will need another registered Internet address uniquely for the Firewall.
- **Translate Secured IP Addresses** - A translate secured IP address entry defines a set of secured network addresses that require NAT to perform IP address translation. By default, the network address translator performs address translation on all secured IP addresses.
- **Exclude Secured IP Addresses** - An exclude secured IP address entry defines a set of secured network addresses that does not require NAT to perform IP address translation. By default, the network address translator performs address translation on all secured IP addresses unless the address is within the range specified by an exclude secured IP addresses entry.
- **MAP Secured IP Address** - A map secured IP address entry defines a one-to-one mapping from a secured IP address to a registered IP address. This one-to-one IP address mapping allows external application clients, such as FTP or Telnet clients, to set up TCP sessions with server machines that reside within the secured network.

The syntax of the NAT command follows:

```
fwnat cmd=list | update | verify | shutdown | startlog | stoplog
```

The parameter definitions are:

fwnat cmd=list

Lists current NAT configuration

fwnat cmd=update

Refreshes the NAT engine

fwnat cmd=verify

Syntax-checks the configuration

fwnat cmd=shutdown

Stops all address translation

fwnat cmd=startlog

Starts logging each translated packet

fwnat cmd=stoplog

Stops logging each translated packet

To add a many-to-one entry to the NAT configuration use **type=many-to-one**:

```
fwnat cmd=add
      type=many-to-one
      addr=Addr
      [timeout=minutes]
```

The parameter definitions are:

type=many-to-one

Adds a many-to-one entry

addr=Addr

IP address that identifies a range of registered IP addresses added to the registered address pool

timeout=minutes

The number of minutes an address translation can remain idle before NAT can free the registered IP address. The default is 15 and the range is 5–45.

To modify a many-to-one entry in the NAT configuration use the following syntax:

```
fwnat cmd=change
      index=
      [addr=Addr]
      [timeout=minutes]
```

The parameter definitions are:

index When you execute `fwnat cmd=list`, there are numbers in the left-hand column for specific NAT entries. Use the number for your specific NAT entry for the index parameter.

addr=Addr

IP address that identifies a range of registered IP addresses added to the registered address pool

timeout=minutes

the number of minutes an address translation can remain idle before NAT can free the registered IP address. The default is 15 and the range is 5–45.

To add a translate entry to the NAT configuration file use **type=translate** and to exclude an entry from the NAT configuration file use **type=exclude**:

```
fwnat cmd=add
      type={translate|exclude}
      addr=Addr
      mask=Mask
```

The parameter definitions are:

type=translate

Adds a translate entry

type=exclude

Adds an exclude entry

addr=Addr

IP address that identifies a range of secured IP addresses that require translation.

mask=Mask

Identifies a range of IP addresses

To modify a translate or exclude entry in the NAT configuration file use the following syntax:

```
fwnat cmd=change
      index=
      [addr=Addr]
      [mask=Mask]
```

The parameter definitions are:

index When you execute `fwnat cmd=list`, there are numbers in the left-hand column for specific NAT entries. Use the number for your specific NAT entry for the index parameter.

addr=Addr
IP address that identifies a range of secured IP addresses that require translation.

mask=Mask
Identifies a range of IP addresses

To add a map entry to the NAT configuration use **type=map**:

```
fwnat cmd=add
      type=map
      secaddr=SecureAddr]
      remaddr=RegisteredAddr]
```

The parameter definitions are:

type=map
Adds a map entry

secaddr
IP address that should be translated into a specified registered address

remaddr
Registered address into which the specified secure address should be translated

To modify a map entry in the NAT configuration use the following syntax:

```
fwnat cmd=change
      index=
      [secaddr=SecureAddr]
      [remaddr=RegisteredAddr]
```

The parameter definitions are:

index When you execute `fwnat cmd=list`, there are numbers in the left-hand column for specific NAT entries. Use the number for your specific NAT entry for the index parameter.

secaddr
IP address that should be translated into a specified registered address

remaddr
Registered address into which the specified secure address should be translated

Network Objects

The `fwnwobj` command is used to create, modify, delete, and list network objects, which are used as endpoints in connections or as members of network object groups. See the *IBM eNetwork Firewall User's Guide* for a complete description of network objects.

To create a network object, issue the following command.

```
fwnwobj cmd=add
      name=name
      [desc=description]
```

```
type={Host|Network|Firewall|Router|Interface|VPN|User}
[addr=x.x.x.x]
[mask=x.x.x.x]
[userid=userid]
```

The parameter definitions are:

- name** Is the name you want to give this network object.
- desc** Is a description of this network object.
- type** Is the type of object that you want to create.
- addr** Is the IP address for this network object. This parameter is not valid if *type=user*.
- mask** Is a mask to indicate how much of the IP address is used in comparisons. This parameter is not valid if *type=user*.
- userid** Is a valid username. This parameter is only valid if **type=user**.

To modify a network object, issue the following command.

```
fwnwobj cmd=change
{name=name | id=id}
[desc=[description]]
[type={Host|Network|Firewall|Router|Interface|VPN|User}]
[addr=x.x.x.x]
[mask=x.x.x.x]
[userid=userid]
```

The parameter definitions are:

- name** Is the name of the network object you want to change.
- id** Is the ID number of the network object you want to change.
- desc** Is a description of this network object.
- type** Is the type of object.
- addr** Is the IP address for this network object. This parameter is not valid if *type=user*.
- mask** Is a mask to indicate how much of the IP address is used in comparisons. This parameter is not valid if *type=user*.
- userid** Is a valid username. This parameter is only valid if **type=user**.

To delete a network object, issue the following command.

```
fwnwobj cmd=delete
{name=name | id=id}
[force={yes|no}]
```

The parameter definitions are:

- name** Is the name of the network object you want to delete.
- id** Is the ID number of the network object you want to delete.
- force** Indicates whether or not this object should be deleted even if other objects are dependent on it.

To list a network object, issue the following command.

```
fwnwobj cmd=list
[name=name] | [id=id]
[format={short|long|wide|raw}]
```


The parameter definitions are:

name Is the name of the network object you want to list.

id Is the ID number of the network object you want to list.

format
Describes the format of this command's output.

Note: If you do not supply a name or an ID, then all network objects will be listed.

Network Object Groups

The `fwnwgrp` command is used to create, modify, delete, and list network object groups. See the *IBM eNetwork Firewall User's Guide* for a complete description of network object groups.

To create a network object group, issue the following command.

```
fwnwgrp cmd=create
      name=name
      [desc=description]
      [idlist=id1,id2,...,idn]
      [namelist="name 1|name2|...|name n"]
```

The parameter definitions are:

name Is the name you want to give to this network object group.

desc Is a description of this network object group.

idlist Is a list of network objects by ID number that you want to have in the network object group.

namelist
Is a list of network objects by name that you want to have in the network object group.

To modify a network object group, issue the following command.

```
fwnwgrp cmd=change
      {name=name | id=id}
      [desc=[description]]
      [idlist=[id1,id2,...,idn]]
      [namelist=["name 1|name2|...|name n"]]
```

The parameter definitions are:

name Is the name of the network object group that you want to change.

id Is the ID number of the network object group that you want to change.

desc Is a description of this network object group.

idlist Is a list of network objects by ID number that you want to have in the network object group.

namelist
Is a list of network objects by name that you want to have in the network object group.

To delete a network object group, issue the following command.

```
fwnwgrp cmd=delete
{name=name | id=id}
[force={yes|no}]
```

The parameter definitions are:

- name** Is the name of the network object group that you want to delete.
- id** Is the ID number of the network object group that you want to delete.
- force** Indicates whether or not this group should be deleted even if other objects are dependent on it.

To list a network object group, issue the following command.

```
fwnwgrp cmd=list
{name=name | id=id}
[format={short|long|wide|raw|all}]
```

The parameter definitions are:

- name** Is the name of the network object group that you want to list.
- id** Is the ID number of the network object group that you want to list.
- format** Describes the format of the command's output.

To add network objects to a network object group, issue the following command.

```
fwnwgrp cmd=add
{name=name | id=id}
{idlist=id1,id2,...,idn| and/or
namelist="name 1|name2|...|name n"}
```

The parameter definitions are:

- name** Is the name of the network object group that you want to change.
- id** Is the ID number of the network object group that you want to change.
- idlist** Is a list of network objects by ID number that you want to add to the network object group.
- namelist** Is a list of network objects by name that you want to add to the network object group.

To remove network objects from a network object group, issue the following command.

```
fwnwgrp cmd=remove
{name=name | id=id}
{idlist=id1,id2,...,idn| and/or
namelist="name 1|name2|...|name n"}
```

The parameter definitions are:

- name** Is the name of the network object group that you want to change.
- id** Is the ID number of the network object group that you want to change.
- idlist** Is a list of network objects by ID number that you want to remove from the network object group.

namelist

Is a list of network objects by name that you want to remove from the network object group.

Paging

You can activate pager notification support to have the firewall page a system administrator by sending a message to the administrator's beeper when there are intrusion alerts on the firewall. For this to work properly, you must configure the pager, the carrier service, and a modem using the `fwpgr`, `fwcarrier`, and the `fwmodem` commands.

Pager Configuration

The `fwpgr` command sets up parameters for your active pager, the one that the Firewall will signal.

To list a pager, issue the following command.

```
fwpgr cmd=list
```

To add a pager, issue the following command.

```
fwpgr cmd=add
    carrier=
    modem=
    type=
    priority=
    pagerid=
    message=
```

To modify pager parameters, issue the following command.

```
fwpgr cmd=change
    [carrier=]
    [modem=]
    [type=]
    [priority]
    [pagerid=]
    [message=]
```

The parameter definitions are:

carrier

A name for the carrier service, as defined in the carriers database (through the `fwcarrier` command).

modem

The same value as the filename parameter of the `fwmodem` command.

type Specify either numeric or alpha.

priority

Enter a priority for sending a page. The highest priority is 5 and the lowest priority is -1.

pagerid

The carrier-assigned, unique identifying number or name for your paging device.

message

The message to be sent to and displayed on the paging device. Either a

number or text, depending on the service your carrier is providing. It will be truncated if it exceeds the smaller of the length setting for the carrier or 200 characters.

Carrier

Use the `fwcarrier` command to set up parameters for any paging services you use.

To list a carrier, issue the following command.

```
fwcarrier cmd=list
        carrier=
```

To add a carrier, issue the following command.

```
fwcarrier cmd=add
        carrier=
        dial=
        nid=
        dtmf=
        digits=
        blocks=
        trans=
        length=
        baud=
        parity=
        databits=
        stopbits=
```

To modify carrier parameters, issue the following command.

```
fwcarrier cmd=change
        carrier=
        [dial=]
        [nid=]
        [dtmf]
        [digits=]
        [blocks=]
        [trans=]
        [length=]
        [baud]
        [parity=]
        [databits=]
        [stopbits=]
```

To delete a carrier, issue the following command.

```
fwcarrier cmd=delete
        carrier=
```

The parameter definitions are:

carrier

The name of the carrier.

dial Must specify the carrier's modem phone number for the TAP service for which you have contracted.

nid Determines whether or not the paging carrier allows numeric IDs to be used during a data connection. Specify Yes for numeric pagers or No for alphanumeric pagers.

digits Specify the maximum number of digits for the alphanumeric pager.

blocks

Specify the maximum blocks per transaction.

- trans** Specify the maximum transactions per call.
- length** The maximum message length permitted by your carrier's service.
- baud** Specify the most reliable baud rate supported by your carrier's service.
- parity** The type of parity checking supported by your carrier's service. This is usually even parity for the TAP protocol.
- databits**
The number of data bits supported by your carrier's service. This is usually 7 for the TAP protocol.
- stopbits**
The number of stop bits supported by your carrier's service. This is usually 1 for the TAP protocol.

Modem Configuration

To set up pager notification support, you need to configure your modem.

Use the modem command to configure a modem for sending pager requests to your pager carrier.

To list a modem, issue the following command.

```
fwmodem cmd=list
      filename=
```

To add a modem, issue the following command.

```
fwmodem cmd=add
      modem=
      filename=
      cmdstring=
      cmdterm=
      dialcmd=
      dialpause=
      dialpound=
      dialstar=
      return=
      cmdresp=
      connresp=
      localecho=
      baud=
      databits=
      stopbits=
      parity=
      default=
      initstring=
      outsideline=
      hangup=
```

To modify modem parameters, issue the following command.

```
fwmodem cmd=change
      modem=
      filename=
      [cmdstring=]
      [cmdterm=]
      [dialcmd=]
      [dialpause=]
      [dialpound=]
      [dialstar=]
      [return=]
      [cmdresp]
```

```
[connresp=]
[localecho=]
[baud=]
[databits=]
[stopbits=]
[parity=]
[default=]
[initstring=]
[outside=]
[hangup=]
```

To delete a modem, issue the following command.

```
fwmodem cmd=delete
filename=
```

The parameter definitions are:

modem

A name for the modem file.

filename

Specify the modem filename. This must end with a .modem extension.

cmdstring

Specify the command mode string. The command mode string contains the set of characters that should be sent while in connect mode. This forces the modem into command mode without hanging up.

cmdterm

Specify the command terminator. The command terminator indicates the character that should be appended to the end of all command sequences to force the modem to accept the command.

dialcmd

Specify the dial command. This is the command sent to the modem in command mode. ATDT works for most modems.

dialpause

Specify the character used in a dial string to force your modem to wait for a short period of time (about 1 second) before continuing with the dial string. This is normally a comma.

dialpound

Specify the character used in a dial string to force your modem to dial the touch tone corresponding to the # sign. This is normally the pound (#) sign.

dialstar

Specify the character used in a dial string to force your modem to dial the touch tone corresponding to the * sign. This is normally the asterisk (*) sign.

return Specify the character to append the dial string in order to force the modem back into command mode after completing the dial string. The semicolon works with most modems.

cmdresp

Specify the string that allows your modem to accept commands. Normally, OK works.

connresp

Specify the string that your modem will output when your carrier has been detected and a connection has been made. Most modems use CONNECT.

localecho

Specify Yes if you want the modem to echo local characters while in connect mode.

baud Specify the baud rate for the modem.

databits

Specify the data bits for the modem.

stopbits

Specify the stop bits for the modem.

parity Specify the parity for the modem.

default

Specify the default device for the modem. This device file must exist under the /dev directory and should match with your configured serial port.

initstring

The initialization string for the modem. Parameters in the string must be suitable for an AT modem command, but the AT should not be included as part of the string. Parameters specified should be coordinated with the communications requirements of your carrier's modem.

outsideline

The number to dial to get an outside line.

hangup

Specify the command to force your modem to hang-up after dialing. The default that works well with most modems is ATH0.

Testing Pager Configuration

To ensure that you have correctly configured your active pager, you can type `fwsendpage` at a command prompt.

The parameter definitions are identical to those for the `fwpgr` command.

Multiple Pagers

If you have need to regularly change your active pager, do the following:

- Make sure you have defined all the needed carriers and modems
- Use `fwpgr` or the configuration client to define and save a pager configuration
- Copy the `etc/config/fwcust.pager` file, giving it a name you can recognize
- Define another pager configuration and copy it and so on until you have copies of all the `fwcust.pager` files you need
- Copy the configuration file you want to activate back to `etc/config/fwcust.pager`

If you are trying to handle shift changes, set up an AIX cron job to automatically repeat the last bullet at the start of each shift.

Security Policies

The `fwsecpolicy` command modifies and lists the defined security policy options.

To list the defined security policy options, issue the following command.

```
fwsecpolicy cmd=list
```

To modify the defined security policy options, issue the following command.

```
fwsecpolicy cmd=change  
[servicelist=[id1,id2,...,idn]]
```

Note that if `cmd=change` is the parameter specified, the command will prompt the user for each of the security policy options.

The DNS security policy for this firewall:

id=20 Permit DNS queries.

id=21 Permit DNS zone transfers.

The logging security policy for this firewall:

id=23 Deny broadcast message to nonsecure interface.

The Socks security policy for this firewall:

id=34 Deny Socks to nonsecure interface.

The temporary security policy for this firewall:

id=26 Shutdown service interface.

id=25 Test IP Routing (debug only).

Services

The `fwservice` command allows you to create, modify, delete, and list services, which are collections of filter rules that implement a protocol. See the *IBM eNetwork Firewall User's Guide* for more information about setting up services.

To create a service, issue the following command.

```
fwservice cmd=create  
name=name  
[desc=description]  
[rulelist=id1/{f|b},id2/{f|b},...,idn/{f|b}]  
[log={yes|no}]  
[fragment={yes|no|only|headers}]  
[tunnel=tunnel_id]  
[time=hhmm-hhmm]          (hh = 00-23, mm = 00-59)  
[month=mmm[-mmm]]        (mmm = Jan, Feb, ..., Dec)  
[day=nn[-nn]]            (nn = 1, 2, ..., 29 or 30 or 31)  
[weekday=ddd[-ddd]]      (ddd = Sun, Mon, ..., Sat)  
[timefilter={activate|deactivate}]
```

The parameter definitions are:

name Is the name you want to give to this service.

desc Is the description of the service.

rulelist

Is a list of rules by ID number and flow direction for this service. *f* indicates a flow of left to right and *b* indicates a flow of right to left.

log Indicates whether logging will occur for this service.

fragment

Indicates the type of fragment control for this service.

tunnel Indicates a tunnel id number for this service.

time Indicates a timeframe by hour for this service to be active or inactive.

month

Indicates a timeframe by month for this service to be active or inactive.

day Indicates a timeframe by day of the month for this service to be active or inactive.

weekday

Indicates a timeframe by weekday for this service to be active or inactive.

timefilter

Indicates whether this service should be active or inactive during the specified timeframe.

To modify a service, issue the following command.

```
fwservice cmd=change
  {name=name | id=id}
  [desc=[description]]
  [rulelist=[id1/{f|b},id2/{f|b},...,idn/{f|b}]]
  [log=[{yes|no}]]
  [fragment=[{yes|no|only|headers}]]
  [tunnel=[tunnel id]]
  [time=hhmm-hhmm] (hh = 00-23, mm = 00-59)
  [month=mmm[-mmm]] (mmm = Jan, Feb, ..., Dec)
  [day=nn[-nn]] (nn = 1, 2, ..., 29 or 30 or 31)
  [weekday=ddd[-ddd]] (ddd = Sun, Mon, ..., Sat)
  [timefilter=[{activate|deactivate}]]
```

The parameter definitions are:

name Is the name of the service you want to change.

id Is the ID number of the service you want to change.

desc Is the description of the service.

rulelist

Is a list of rules by ID number and flow direction for this service. *f* indicates a flow of left to right and *b* indicates a flow of right to left.

log Indicates whether logging will occur for this service.

fragment

Indicates the type of fragment control for this service.

tunnel Indicates a tunnel id number for this service.

time Indicates a timeframe by hour for this service to be active or inactive.

month

Indicates a timeframe by month for this service to be active or inactive.

day Indicates a timeframe by day of the month for this service to be active or inactive.

weekday

Indicates a timeframe by weekday for this service to be active or inactive.

timefilter

Indicates whether this service should be active or inactive during the specified timeframe.

To delete a service, issue the following command.

```
fwservice cmd=delete
      {name=name | id=id}
      [force={yes|no}]
```

The parameter definitions are:

name Is the name of the service you want to delete.

id Is the ID number of the service you want to delete.

force Indicates whether or not this service should be deleted even if other objects are dependent upon it.

To list a service, issue the following command.

```
fwservice cmd=list
      [name=name | id=id]
      [format={short|long|wide|raw}]
```

The parameter definitions are:

name Is the name of the service you want to list.

id Is the ID number of the service you want to list.

format

Describes the format of the command's output.

To add a list of rules to a service, issue the following command.

```
fwservice cmd=add
      rulelist=id1/{f|b},id2/{f|b},...,idn/{f|b}
      service=service_id_or_name
```

The parameter definitions are:

rulelist

Is a list of rules by ID number and flow direction to add to this service. *f* indicates a flow of left to right and *b* indicates a flow of right to left.

service

Is the name or ID number of the service you want to change.

To remove a list of rules to a service, issue the following command.

```
fwservice cmd=remove
      rulelist=id1/{f|b},id2/{f|b},...,idn/{f|b}
      service=service_id_or_name
```

The parameter definitions are:

rulelist

Is a list of rules by ID number and flow direction to remove from this service. *f* indicates a flow of left to right and *b* indicates a flow of right to left.

service

Is the name or ID number of the service you want to change.

To move a rule within a service, issue the following command.

```
fwservice cmd=move
      rule=id/{f|b}
      after=[id/{f|b}]
      service=service_id_or_name
```

The parameter definitions are:

rule Is the ID number and flow direction for the rule you want to move in this service.

after Is the ID number and flow direction of the rule you want the moved rule to follow.

service

Is the name or ID number of the service you want to change.

Socks Rules

The `fwsrule` command is used to create, modify, delete, and list Socks rules. See the *IBM eNetwork Firewall User's Guide* for a complete explanation of the Socks rules concepts.

To add a socks rule, issue the following command.

```
fwsrule cmd=add
      name=name
      [desc=description]
      type={permit|deny}
      [identd={None|?=I|?=i|?=n}]
      [userid=userid1,userid2,...,useridn]
      [operator={eq|neq|lt|gt|le|ge}]
      [port=port_num] (port_num=0-65535)
      [rulecommand="rule command"]
```

The parameter definitions are:

name Is the name you want to give this Socks rule.

desc Is the description of the Socks rule.

type Indicates if this rule will permit or deny traffic.

identd Indicates how to use the results of user identification through `identd`.

userid Is a list of users on the requesting host.

operator

Indicates the logical operation to be performed on the port number.

port Indicates a port number (0–65535).

rulecommand

A command string to be executed when the rule conditions are satisfied.

To modify a socks rule, issue the following command.

```
fwsrule cmd=change
      {name=name | id=id}
      [desc=[description]]
      [type={permit|deny}]
      [identd=[{None|?=I|?=i|?=n}]]
```

```
[userid=[userid1,userid2,...,useridn]]
[operator=[{eq|neq|lt|gt|le|ge}]]
[port=[port_num]] (port_num=0-65535)
[rulecommand=["rule command"]]
```

The parameter definitions are:

- name** Is the name of the Socks rule you want to change.
- id** Is the ID number of the Socks rule you want to change.
- desc** Is the description of the Socks rule.
- type** Indicates if this rule will permit or deny traffic.
- identd** Indicates how to use the results of user identification through identd.
- userid** Is a list of users on the requesting host.
- operator**
Indicates the logical operation to be performed on the port number.
- port** Indicates a port number (0–65535).
- rulecommand**
A command string to be executed when the rule conditions are satisfied.

To delete a socks rule, issue the following command.

```
fwsrule cmd=delete
      {name=name | id=id}
      [force={yes|no}]
```

The parameter definitions are:

- name** Is the name of the Socks rule you want to delete.
- id** Is the ID number of the Socks rule you want to delete.
- force** Indicates whether or not this rule should be deleted even if other objects are dependent upon it.

To list a socks rule, issue the following command.

```
fwsrule cmd=list
      [{name=name | id=id}]
      [format={short|long|wide|raw}]
```

The parameter definitions are:

- name** Is the name of the Socks rule you want to list.
- id** Is the ID number of the Socks rule you want to list.
- format**
Describes the format of this comment's output.

Tunnels

The `fwtunnl` command is used to list, add, change, delete, import, export, activate, deactivate, and shutdown tunnels. Three types of tunnel definitions are supported: IBM Tunnels, Manual Tunnels, and Dynamic Tunnels. See the *IBM eNetwork Firewall User's Guide* for a complete explanation of tunnel parameters.

To display all tunnels from the tunnel context files, issue the following command.

```
fwtunn1 cmd=list
[directory=AbsolutePath]
[tunnel={tunnel_id1,tunnel_id2,...,tunnel_idn}]
```

An example of the output generated by the list command is:

```
123 Inactive IBM 9.37.53.88 9.37.54.52 CDMF ae
534 Active IBM 10.100.100.1 9.37.54.52 CDMF ae
679 Inactive manual 10.100.100.1 9.37.61.107 CDMF ae
368 dynamic 9.37.53.88 destination CDMF ae 480
```

To add a tunnel definition to the tunnel context files, for tunnel **type=IBM**, issue the following command.

```
fwtunn1 cmd=add
type=IBM
addr=LocalIPAddress
remaddr=TargetIPAddress
policy={be|ae|encr|auth|none}
encrypthow={DES_CBC_4|DES_CBC_8|CDMF}
tunnel=TunnelID
timeout={1-1440}
refresh={1-720}
algorithm={KEYED_MD5|HMAC_MD5}
initiator={y|n}
```

To modify a specified tunnel, for tunnel **type=IBM**, issue the following command.

```
fwtunn1 cmd=change
tunnel=TunnelID
[addr=LocalIPAddress]
[remaddr=TargetIPAddress]
[policy={be|ae|encr|auth|none}]
[encrypthow={DES_CBC_4|DES_CBC_8|CDMF}]
[timeout={1-1440}]
[refresh={1-720}]
[algorithm={KEYED_MD5|HMAC_MD5}]
[initiator={y|n}]
```

To add a tunnel definition to the tunnel context files, for tunnel **type>manual** and **type=dynamic**, issue the following command.

```
fwtunn1 cmd=add
{type>manual remaddr= | type=dynamic username=}
addr=LocalIPAddress
policy={be|ae|encr|auth|none}
encrypthow={DES_CBC_4|DES_CBC_8|CDMF}
tunnel=TunnelID
timeout={1-99999}
spi={1-999999}
algorithm={KEYED_MD5|HMAC_MD5}
```

To modify a specified tunnel, for tunnel **type>manual** and **type=dynamic**, issue the following command.

```
fwtunn1 cmd=change
tunnel=TunnelID
[remaddr=]
[username=]
[addr=]
[policy={be|ae|encr|auth|none}]
[encrypthow={DES_CBC_4|DES_CBC_8|CDMF}]
[timeout={1-99999}]
[spi={1-999999}]
[algorithm={KEYED_MD5|HMAC_MD5}]
```

To delete a specified tunnel, issue the following command.

```
fwtunnel cmd=delete
        tunnel={tunnel_id1,tunnel_id2,...,tunnel_idn}
```

To import tunnel context files into the given directory, issue the following command. The context files include `fwexpmctx.manual`, `fwexpolicy`, and `fwexpolicy.3.1`.

```
fwtunnel cmd=import
        directory=AbsolutePath
        tunnel={tunnel_id1,tunnel_id2,...,tunnel_idn}
```

To export tunnel context files (for one tunnel only) into the given directory, issue the following command. The context files include `fwexpmctx.manual`, `fwexpolicy`, and `fwexpolicy.3.1`.

```
fwtunnel cmd=export
        directory=AbsolutePath
        tunnel={tunnel_id1,tunnel_id2,...,tunnel_idn}
```

The directory given must not contain existing context files.

To activate or deactivate the specified tunnel, issue the following command.

```
fwtunnel cmd={activate|deactivate}
        [tunnel={tunnel_id1,tunnel_id2,...,tunnel_idn}]
```

If *tunnel* is not specified, the firewall will attempt to activate or deactivate all tunnels and terminate upon the first failure. When activating a set of tunnels, all tunnels in the set must be inactive. Similarly, when deactivating a set of tunnels, all tunnels in the set must be active.

To shutdown all active tunnels, issue the following command.

```
fwtunnel cmd=shutdown
```

The parameter definitions are:

type=IBM Tunnel

A tunnel between two IBM Firewalls. Features an automatic key refresh mechanism.

type=Manual Tunnel

A tunnel between an IBM Firewall and a non-IBM Firewall. Uses the IPSec standard.

type=Dynamic Tunnel

A special tunnel for use with the secure remote client function of the firewall. This tunnel is activated when a remote client initiates a session with the firewall and deactivates when the client is finished.

addr In the local address field, enter the IP address of the local firewall nonsecure interface to be used by the tunnel. You can choose to click Select and select a network object that has already been defined.

remaddr

The target IP address. IP address of the partner firewall to be used by the tunnel.

policy Allows you to enter a combination of encryption and authentication values. Depending upon your security requirements for this tunnel, choose from among the following policies:

encr/auth

Data is encrypted and then authenticated.

auth/encr

Data is authenticated and then encrypted.

encr only

Data is encrypted only.

auth only

Data is authenticated only.

none Data is neither encrypted nor authenticated.

encrypthow

Algorithm used for IP packet encryption. If used, must specify either DES_CBC (for domestic USA or Canada only) or Commercial Data Masking Facility (CDMF), which is used world wide. DES_CBC_8 uses a 64 bit initialization vector and DES_CBC_4 uses a 32 bit initialization vector.

tunnel Enter the tunnel identification number for the tunnel. Valid values are from 1 through 999999.

timeout

Tunnel expiration in minutes.

refresh

Valid for IBM tunnels only. Enter the time in minutes. This determines the amount of overlap time between a new key start and an old key expiration. The value that you specify will affect performance (the smaller the value, the bigger the performance hit). A recommended value is 480 minutes (8 hours).

initiator

The initiator field identifies which partner starts the session negotiations. If both partners are identified as the initiator, the tunnel logic will resolve the deadlock. You must set at least one of the partners as the initiator. Choose yes or no.

spi

The Target Security Parameter Index is used when you add a manual tunnel. This index should be generated by the target side of the tunnel. Use Target SPI and the destination IP address to determine which security association to use. Valid values are from 1 to 999999. However, be aware that values 1 through 255 are reserved for the Internet Assigned Numbers Authority (IANA) for future use.

directory

Specifies location of tunnel context files.

username

Tunnel manipulations require a username for authentication.

Users

This command adds a new user or modifies one or more attributes of an existing firewall user. All parameters either have default values or are unnecessary in certain circumstances. For cmd=add, default values will be stored; for cmd=change, the existing values will be preserved.

```
fwuser cmd={add|change}
      username=LoginName
      [fullname="UsersRealName"]
      [password={yes|no}]
      [pwdvalue=Password]
      [level={proxy|admin}]
      [secshell=SecureShell]
```

```

[remshell=NonSecureShell]
[locllogin=LocalLoginAuthentication]
[secftp=SecureFTPAuthentication]
[remftp=NonSecureFTPAuthentication]
[secauth=SecureTelnetAuthentication]
[remauth=NonSecureTelnetAuthentication]
[remip=NonSecureIPSecClientAuthentication]
[secadmin=SecureAdminAuthentication]
[remadmin=NonSecureAdminAuthentication]
[warntime=IdleWarningTime]
[disctime=IdleDisconnectTime]
[histexpire=HistoryExpiration]
[histsize=HistorySize]
[loginretries=LoginRetries]
[maxage=MaxAge]
[maxexpired=MaxExpiredAge]
[maxrepeats=MaxRepeatChars]
[minalpha=MinAlphaChars]
[mindiff=MinDifferentChars]
[minlen=MinLength]
[mianother=MinNonAlphaChars]
[pwdwarntime>PasswordWarnTime]
[modeallowed=host|none]
[fg_act={yes|no}]
[fg_all={yes|no}]
[fg_addrtrans={yes|no}]
[fg_clone={yes|no}]
[fg_dist={yes|no}]
[fg_dns={yes|no}]
[fg_interfaces={yes|no}]
[fg_logmonitor={yes|no}]
[fg_logs={yes|no}]
[fg_mail={yes|no}]
[fg_netobjs1={yes|no}]
[fg_netobjs2={yes|no}]
[fg_pagers={yes|no}]
[fg_proxyserver={yes|no}]
[fg_secag={yes|no}]
[fg_sesslfm={yes|no}]
[fg_snmp={yes|no}]
[fg_user={yes|no}]
[fg_traffic={yes|no}]
[fg_vpn={yes|no}]

```

Fundamental Parameters

username

Login name for this user. Must be a valid AIX login name.

fullname

User's full name, or some other brief (one-line) information pertaining to this user. If spaces are to be included in this value, the value must be enclosed in double-quotes.

level The default value is proxy, which indicates that the user being created is a simple proxy or Socks user. Administration function groups and administration authentications do not apply to proxy users.

Login Shells

secshell

Shell to use for telnet logins from a secure interface. Valid values are /bin/restrict.sh, /bin/csh, /bin/ksh, /bin/bsh, and /bin/oneact.sh. The default is /bin/restrict.sh.

remshell

Shell to use for telnet logins from a nonsecure interface. Valid values are /bin/restrict.sh, /bin/csh, /bin/ksh, /bin/bsh, and /bin/oneact.sh. The default is /bin/restrict.sh.

Authentications

Following are authentication strings and their corresponding authentication methods. Use of the authentication strings for the various parameters of the fwuser command is indicated below.

- permit—permit all
- deny—deny all
- password—Firewall password
- sdi—SDI
- user_defined—user-supplied authentication

loclogin

Authentication method for logins from the local console. Valid values are deny, permit, password, sdi, and user_defined. The default is deny.

secftp Authentication method for FTP logins from a secure interface. Valid values are deny, permit, password, sdi, and user_defined. The default is deny.

remftp

Authentication method for FTP logins from a nonsecure interface. Valid values are deny, permit, password, sdi, and user_defined. The default is deny.

secauth

Authentication method for telnet logins from a secure interface. Valid values are deny, permit, password, sdi, and user_defined. The default is deny.

remauth

Authentication method for telnet logins from a nonsecure interface. Valid values are deny, permit, password, sdi, and user_defined. The default is deny.

remip Authentication method for Remote IPSec Client logins from a nonsecure interface. Valid values are deny, none, password, sdi, and user_defined. The default is deny.

secadmin

Authentication method for Firewall Configuration Client logins from a secure interface. Valid values are deny, permit, password, sdi, and user_defined. The default is deny.

remadmin

Authentication method for Firewall Configuration Client logins from a nonsecure interface. Valid values are deny, permit, password, sdi, and user_defined. The default is deny.

Note: fwduser cannot have Firewall Password set on any of its authentication method fields.

Idle Proxy Parameters**wartime**

Idle time in minutes after which the fwidleout command will warn this user about a forthcoming disconnection.

disctime

Idle time in minutes after which the `fwidleout` command will disconnect this user. Disctime should be greater than warntime.

Firewall Password Parameters**password**

Indicates if a user will be prompted for a password. By default, you will be prompted if any authentication method is specified or allowed to default to password.

pwdvalue

Used mostly for script programming, this parameter allows the value of a parameter to be specified on the command line. Note that this value is entered in clear text and is not obscured from eavesdroppers. There is no default.

histexpire

Defines the period of time (in weeks) that a user cannot reuse a password. The value is an integer string. The valid values are 0 - 52. The value of 0 indicates no time limit is set. The default value is 0.

histsize

Defines the number of previous passwords a user cannot reuse. The value is an integer string. The valid values are 0 - 20. Only valid if `histexpire=0`. The default value is 5.

loginretries

Defines the number of unsuccessful login attempts allowed after the last successful login before the system locks the account. The value is an integer string. The valid values are 0 - 20. The default value is 10. A zero or negative value indicates that no limit exists. Once the user's account is locked, the user will not be able to log in until the system administrator resets the user's `unsuccessful_login_count` attribute in the `/etc/security/lastlog` file to be less than the value of login retries. To do this, enter the following,

```
chsec -f /etc/security/lastlog -s username -a unsuccessful_login_count=<value>
```

maxage

Defines the maximum age (in weeks) of a password. The password must be changed by this time. The value is an integer string. The valid values are 0 - 52. The value of 0 indicates no maximum age. The default is 13.

maxexpired

Defines the maximum time (in weeks) beyond the `maxage` value that a user can change an expired password. After this defined time, only an administrative user can change the password. The value is an integer string. The valid values are -1 - 26. If the `maxexpired` attribute is 0, the password expires when the `maxage` value is met. If the `maxage` attribute is 0, the `maxexpired` attribute is ignored. The default is 3.

maxrepeats

Defines the maximum number of times a character can be repeated in a new password. The valid values are 0 - 8, but a value of 0 is meaningless. The value of 8 indicates that there is not a maximum number. The default is 2.

minalpha

Defines the minimum number of alphabetic characters that must be in a

new password. The value is an integer string. The valid values are 0 - 8. The value of 0 indicates no minimum number. The default is 4.

mindiff

Defines the minimum number of characters required in a new password that were not in the old password. The value is an integer string. The valid values are 0 - 8. The value of 0 indicates no minimum number. The default is 3.

minlen

Defines the minimum length of a password. The value is an integer string. The valid values are 0 - 8. The value of 0 indicates no minimum number. The default is 8.

minother

Defines the minimum number of non-alphabetic characters that must be in a new password. The value is an integer string. The valid values are 0 - 8. The value of 0 indicates no minimum number. The default is 1.

pwdwarntime

Defines the number of days before the system issues a warning that a password change is required. The value is an integer string. The valid values are 0 - 30. A zero or negative value indicates that no message is issued. The default value is 5.

Administration Functional Groups

modeallowed indicates the login modes allowed:

- none—User is not allowed to login to the firewall configuration server
- efm—Administrator can login in EFM mode
- host—Administrator can login to the firewall configuration server host mode only.
- both—Administrator can login in either EFM mode or Host mode

fg_all Enter yes if this administrator is allowed to administer all aspects of the firewall. The default is no.

fg_act Enter yes if this administrator is allowed to activate changes on a managed firewall. The default is no.

fg_addrtrans

Enter yes if this administrator is allowed to administer Network Address Translation. The default is no.

fg_clone

Enter yes if this administrator is allowed to clone a managed firewall. The default is no.

fg_dist

Enter yes if this administrator is allowed to transmit configuration changes to a managed firewall. The default is no.

fg_dns

Enter yes if this administrator is allowed to administer Domain Name Services. The default is no.

fg_interfaces

Enter yes if this administrator is allowed to define firewall interfaces. The default is no.

fg_logmonitor

Enter yes if this administrator is allowed to administer Log Monitor thresholds. The default is no.

fg_logs

Enter yes if this administrator is allowed to administer Log Facilities. The default is no.

fg_mail

Enter yes if this administrator is allowed to administer the firewall mail gateway. The default is no.

fg_netobjs1

Enter yes if this administrator is allowed to perform basic administration of Network Objects. The default is no.

fg_netobjs2

Enter yes if this administrator is allowed to perform advanced administration of Network Objects. The default is no.

fg_pagers

Enter yes if this administrator is allowed to administer Pager Setup. The default is no.

fg_proxyserver

Enter yes if this administrator is allowed to configure the firewall proxy daemons. The default is no.

fg_secag

Enter yes if this administrator is allowed to administer a managed Firewall's Security Agreement. The default is no.

fg_sesslfn

Enter yes if this administrator is allowed to administer a managed Firewall's session limits. The default is no.

fg_snmp

Enter yes if this administrator is allowed to administer SNMP managers and subagent. The default is no.

fg_traffic

Enter yes if this administrator is allowed to administer Traffic Control. The default is no.

fg_user

Enter yes if this administrator is allowed to administer firewall users. The default is no.

fg_vpn

Enter yes if this administrator is allowed to administer Virtual Private Networks. The default is no.

To list all attributes of all firewall users or of a single specified firewall user:

```
fwuser cmd=list  
      [username=username]  
      [type={short|long}]
```

type={short|long}

The default for type is long if you use a username. If you do not use a username, the default is short.

To remove a user from the firewall:

```
fwuser cmd=delete  
username=username
```

Chapter 2. Using Report Utilities

This chapter discusses using the report utilities of the IBM Firewall. The primary purpose of the report utilities is to generate tabulated files of administrative information from firewall log files.

Tabulated text files can be generated and imported into tables in a database system, such as DB2/6000 or DB2/2. The administrator can then use the Structured Query Language (SQL) to query the data and generate reports. The utilities also allow the administrator to create a readable text file of the firewall log messages.

Report utilities consist of the following programs and files:

fwar2asc

Program to extract firewall log files from an archive library

fwlogtxt

Program to generate full-text messages from a firewall log file

fwlogtbl

Program to generate database import files, in DEL (delimited) format, from a firewall log and an su log.

To use the fwlogtbl program and the DDL, DML, and DEL files, you should have some knowledge of relational databases and the use of an appropriate relational database product.

fwschema.ddl

File of SQL Data Definition Language (DDL) statements, suitable for defining the database tables

fwimport.dat

File of DB2 import statements, suitable for importing the DEL files into the database tables

fwqrysmpl.dml

File of SQL Data Manipulation Language (DML) statements, suitable for generating sample reports

The DDL and DML files are specific to the DB2 family, but can be modified for use with other database management systems. DEL format files can be readily imported (loaded) into DB2/6000, DB2/2, and other database and file systems. Their simple format should allow conversion to other formats, if necessary.

In addition to processing the firewall log file, the administrator can use the utilities to process the AIX su log file (usually **/var/adm/sulog**). This file contains information about attempted uses of the AIX su command. Logged-in users use the AIX su command to switch to a different user ID, potentially acquiring greater authority. Both successful and unsuccessful attempts are logged. The result of processing the su log file is a tabulated file that can be imported into a database system.

Note: Do not use report utilities to process a log from the IBM Firewall for Windows NT. Only use the IBM Firewall for Windows NT report utilities to process a log from the IBM Firewall for Windows NT.

Report Utilities Usage

This information explains how to use report utilities from the command line and SMIT. Refer to the *IBM eNetwork Firewall User's Guide* for information on using the report utilities from the configuration client.

To view the firewall log file from the command line, use the **fwlogtxt** utility. See "Generating Messages from the Firewall Log File" for more information.

To view the firewall log file in SMIT, select the Create Expanded Text Message File option. See "The SMIT Layout Panel" on page 49 for more information.

To generate reports based on log information:

1. Install the relational database product.
2. Create an empty database.
3. Create empty firewall log tables in the database.
4. To produce the tabulated files, run **fwlogtbl** from the command line or select the Create Tabulated Text Manager File from SMIT.
5. Import the resulting files to populate the database tables with log data.
6. Produce reports by running SQL statements or SQL programs.

Note: The first three steps need to be done once, while the remaining steps are repeated each time new log data is available.

IBM Firewall Log Format

Each entry of the firewall log file has the format:

```
Date Time firewall_name:year;pid:msg_num; msg_ID;var_1;...;var_n;
```

where

- The first three fields, **date**, **time**, and **firewall-name** are added by syslog.
- **year** is the four-character year.
- **pid** is the AIX process ID to which the entry applies.
- **msg_num** is a sequential integer which the Report Utilities use to access the appropriate, translated message text from the fw_log.cat file.
- **msg_ID** is the external number of the message (such as ICA0001e).
- **var_1-n** represent the values of message variables, where **n** is the number of variables in the message definition.

Note: Do not direct other syslog records to the same file as the firewall log. Such records will not conform to the format required by the report utilities and results are not predictable.

Generating Messages from the Firewall Log File

Use the command **fwlogtxt** to generate readable messages from the entries of a firewall log file.

The parameters include:

input Standard input from a firewall log file

output
Standard output

fwlogtxt syntax

```
fwlogtxt
```

Example:

```
fwlogtxt < fw980212.log >logtxt.out  
fwlogtxt < fw980212.log | grep ICA31  
tail -f /var/adm/messages | fwlogtxt
```

There are no parameters for **fwlogtxt**; it takes information from the standard input and puts results to the standard output.

Note that the second example filters the output to show full text of only those messages that start with 'ICA31'. Additional filtering can be done using standard AIX facilities or user-provided scripts/programs. The third example of invocation (**tail -f**) permits dynamic monitoring of an active log and could also be filtered.

Extracting Firewall Log Files

Use the command **fwar2asc** to extract the named files from an archive library file into an ASCII file. The resulting ASCII file can be used as an input for both **fwlogtxt** and **fwlogtbl**. The archive library file is assumed to be in the format generated by Log File Management (the **fwlogmgmt** command). That is, the archived logs are compressed and end in '.Z'

fwar2asc syntax

```
fwar2asc [-f OutFile] ArchiveFile LogName
```

Example:

```
fwar2asc -f myFwLog myFwLogs.a 961113fwLog.Z  
fwar2asc myFwLogs.a 961113fwLog.Z
```

-f **-f** is AIX syntax.

OutFile

Specifies the directory and file name of the output ASCII file. The default directory is the current directory. The default file name is the same as the value of the **LogName** parameter, but without terminating '.Z'.

ArchiveFile

Specifies the directory and file name of the input archive library file.

LogName

Specifies the name of the log file the user wants to extract from the archive library file. This will most likely end in '.Z' since the archived logs are compressed.

Generating Database Import Files

Use the command **fwlogtbl** to create, write over, or append to the tabulated files from which the user can populate the database tables for report generation.

The parameters include:

input Firewall log file, for example one extracted from a log archive library file using fwar2asc. -w, -a, and -su are parameter options.

output

File names:

- a_alert.tbl
- f_rule.tbl
- f_info.tbl
- f_match.tbl
- f_stat.tbl
- nat_info.tbl
- p_info.tbl
- p_ftp.tbl
- p_http.tbl
- p_info.tbl
- p_login.tbl
- p_stat.tbl
- server_info.tbl
- session.tbl
- s_ftp.tbl
- s_info.tbl
- ssl_info.tbl
- su.tbl
- t_cntxt.tbl
- t_policy.tbl
- t_stat.tbl

fwlogtbl syntax

```
fwlogtbl -w [-d OutDir] [-su]LogName
          |
          -a
```

Example:

```
fwlogtbl -a -d /u/tai/fw/reports fw961031.log
```

-w Specifies that the existing output file should be replaced. If the file does not exist, fwlogtbl creates it.

-a Specifies that the file generated should be appended to the existing output file. If the file does not exist, fwlogtbl creates it.

-d Identifies the output directory.

OutDir

Specifies the directory in which all the output files are to be stored. If no directory is specified, the output files will be stored in the current directory.

-su Specifies that the LogName is the name of an AIX su log file.

LogName

Specifies a firewall log file or an AIX su log file.

The output file names are predefined but can be copied or renamed after running `fwlogtbl`. The output files have delimited ASCII (DEL) file format, with no character string delimiters, and use semicolon (;) as the column delimiters.

For more information on messages, see “Appendix A. Messages” on page 101.

Using a Database with Report Utilities

This section describes files provided with the firewall for creating the database, importing information into the database, and querying reports. If you have DB2, the `db2` command can be used with these files. (Functions similar to the `db2` command might exist in other database managers. The files may require alteration to be used with such functions.)

To run the `db2` command, you must have DB2 installed and an instance defined. See the DB2 install documentation. Initially, you must use DB2’s create database command to create an empty database. (We suggest calling it *fwlog*). To do this, type at the command line:

```
db2 create database fwlog
```

You must then connect to *fwlog*:

```
db2 connect to fwlog
```

The `-vf` options of the `db2` command can then be used as follows:

```
db2 -vf fwschema.ddl > schema.out
db2 -vf fwimport.dat > import.out
db2 -vf fwqrysmpl.dml > report.out
```

These steps are described in more detail in the following sections. In each case, the user should carefully check the standard output (redirected to a file in each of the examples). For import, it is also necessary to check the `.msg` file produced by each individual import statement.

Creating the Tables

The command **`db2 -vf fwschema.ddl > schema.out`** creates all the tables and indexes needed. Issue this command once, preferably soon after installing the firewall. The current user ID at the time this example is run will be the creator ID of the tables. This ID may need to be used as a table name qualifier (such as `creatorid.tableName`) in later SQL statements, unless they are run under the creator’s ID. Thus, if not using the creator’s ID, the user will need to edit the `fwimport.dat` and `fwqrysmpl.dml` files to place the creator ID in front of each table name.

The `/usr/lpp/FW/sample/fwschema.ddl` file contains the DDL statements to create the database tables needed to accept records from the tabulated files created by `fwlogtbl`. You should look at `schema.out` to determine if your operation was successful. The statements in `fwschema.ddl` file can be used as is or can be modified to work with various database systems. (Users should not change table and column names.)

Importing the Data

The command **`db2 -vf fwimport.dat > import.out`** loads data from all the DEL files into the tables created by the **`db2 -vf fwschema.ddl`** command.

The `/usr/lpp/FW/sample/fwimport.dat` file contains sample statements for importing the data from the `*.tbl` files into the DB2 database. As mentioned in “Creating the Tables” on page 47, if the user of the imports is not the creator of the tables, the creator ID must be placed in front of each table name.

Each import statement produces information in standard out and additional information in a `tblname.msg` file, where `tblname` is specific to each import statement. The user should check both forms of output to determine if the import was successful. When running all the import statements in this file with a program such as DB2, the user should direct standard out to a file, then check that file and each of the `.msg` files. Each one of the import commands produces a separate `.msg` file. Also, the user should re-issue the `db2 -vf fwimport.dat > import.out` command whenever they have a new log to reflect in the database.

When importing large log files you might receive SQL error codes with descriptions indicating the need for more memory or disk space. For example, the message might be insufficient heap space or transaction log space. These errors require adjustment of the parameter settings for the database product or for the `fwlog` database. See the DB2 documentation for more information. A temporary alternative to adjusting the DB2 parameter settings is to split large logs or large tabulated files into smaller files.

Running Sample Queries

The `db2 -vf fwqrysmp.dml > report.out` command runs the sample queries. The `fwqrysmp.dml` file, `/usr/lpp/FW/sample/fwqrysmp.dml`, contains sample SQL statements that can provide useful report data, based on some of the query requirements. You can build on these examples to create your own reports. As mentioned in “Creating the Tables” on page 47, if the user of the imports is not the creator of the tables, the creator ID must be placed in front of each table name.

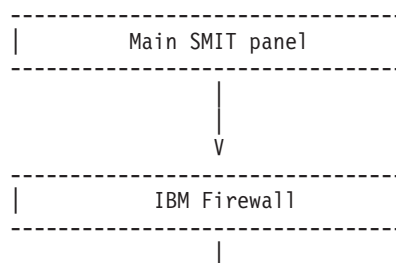
When running queries from the command line, DB2 allocates the maximum space it might need for each output column. This can result in a report that is difficult to read. You might achieve more satisfactory results by requesting fewer columns in each query or by imbedding these query statements in a program where you can better control the presentation.

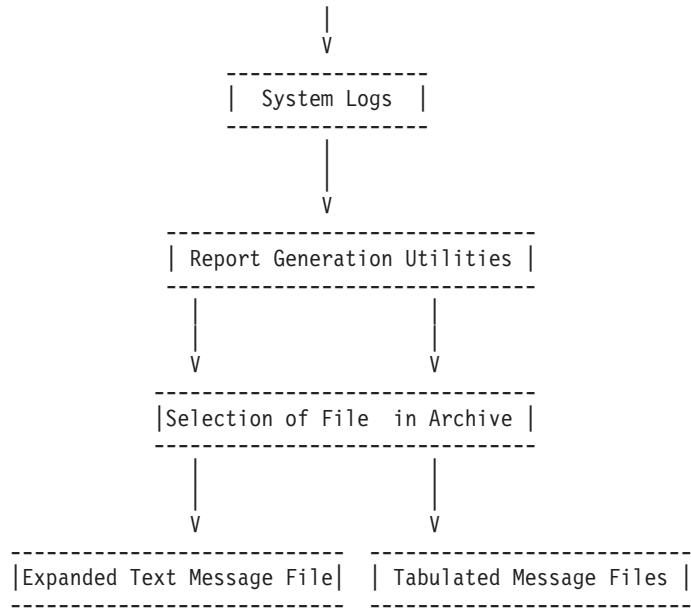
User Interface into Report Utilities

Report Utilities are installed as part of the firewall installation. They can also be separately installed and run on a non-firewall host. The configuration client or the `fwlogtbl` command can be used to run report utilities on the firewall. On a non-firewall machine, you will use SMIT or the command line.

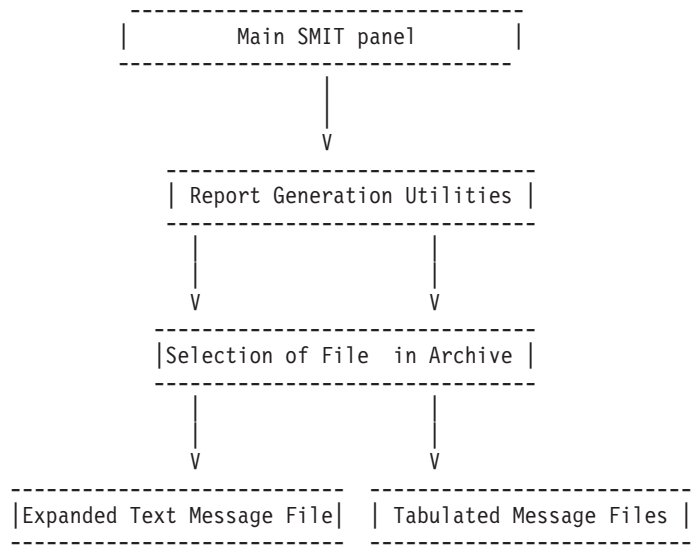
Using the Reports with SMIT

This diagram illustrates the sequence leading to the panels for a firewall machine.





This is the sequence leading to the panels in a non-firewall machine.



The SMIT Layout Panel

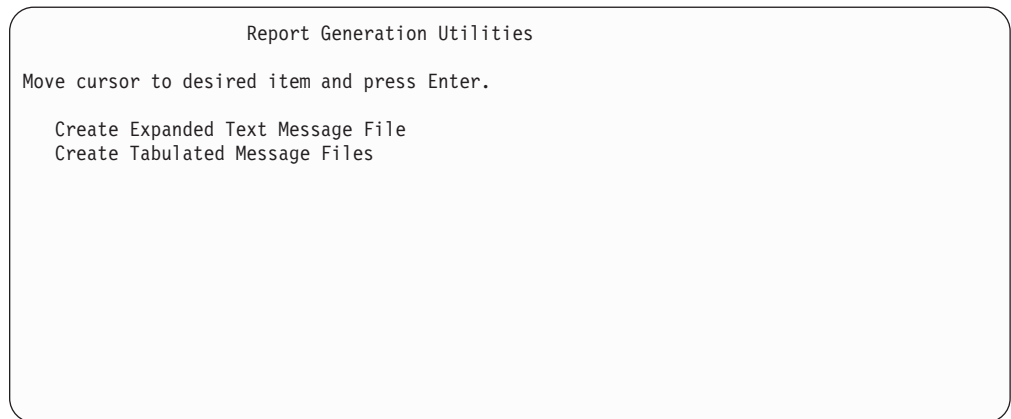


Figure 1. The selection of Report Utility Type

The panel in figure 2 requests the name of a log archive. See figure 3 for usage of this information. A similar panel appears if Create Tabulated Message Files was selected.

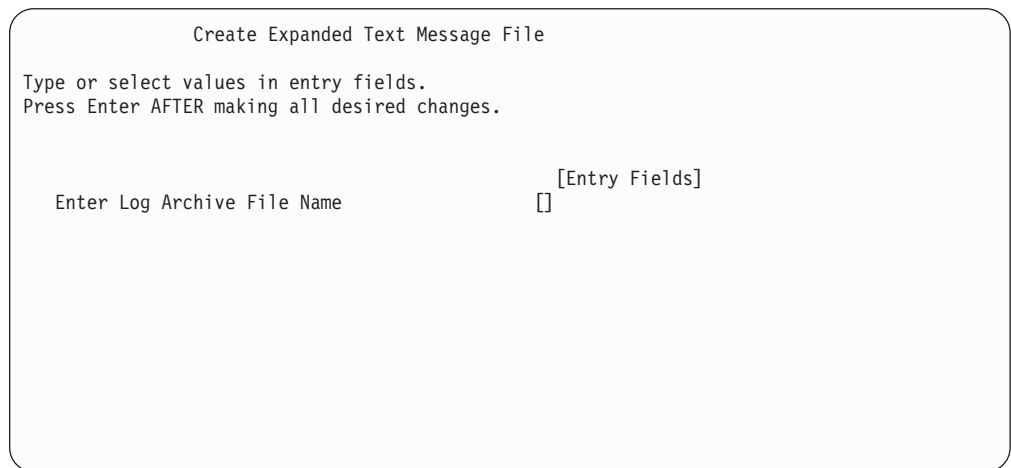


Figure 2. The field for Log Archive File Name entry

The list for Log File Name field will be the list of names extracted from the Log Archive named on the preceding panel and shown at the bottom of this one. The names will end in .Z, since they have been compressed by the fwlogmgmt command. If no Log Archive name was given on the preceding panel, the list will show names of all files in the directory SMIT was started from. It is assumed that these are not compressed files. The Message Filter is used to select a subset of the log messages (which have an ICA prefix). The default filter is all messages in the log (ICA). A filter of ICA3 would select all messages related to Socks. A filter of ICA3012 would show the text of only that one message. The Default Output File is standard output.

For more information, see "Appendix A. Messages" on page 101.

```

Create Expanded Text Message File

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

Log File Name          [Entry Fields]
Message Filter         []+
Path and File Name for Output Text []
Log Archive File Name  []

```

Figure 3. The fields for "Create Expanded Text Message File"

See the description of the Create Expanded Text Message File panel for details about the list for Log File Name field. The Default for Directory for Output Files is the current directory

```

Create Tabulated Message Files

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

Log File Name          [Entry Fields]
Log File Type          []+
Append to existing files [Firewall log]+
Directory for Output Files [yes]+
Log Archive File Name  []

```

Figure 4. The fields for "Create Tabulated Message Files"

The SQL Tables

This section defines the layout of the SQL tables.

Each firewall log message or AIX su log message is mapped to one of the following SQL tables:

```
ADMIN_ALERT
FILTER_INFO
FILTER_MATCH
FILTER_ACTIVE_RULE
FILTER_STATUS
NAT_INFO
PAGER_INFO
PROXY_FTP
PROXY_HTTP
PROXY_INFO
PROXY_LOGIN
PROXY_STATUS
SERVER_INFO
SESSION
SOCKS_FTP
SOCKS_INFO
SSL_INFO
SU
TUNNEL_CONTEXT
TUNNEL_POLICY
TUNNEL_STATUS
```

You should not change the table and column names. However, you can increase the width of a char column if you find that some of its values are being truncated.

Indexes

A log record representing a particular firewall event should appear only once in the database. If an administrator imports the same tabulated file multiple times or if another tabulated file derived from the same log file is imported, a log record could appear more than once.

To help avoid this problem, the database definition sample file, fwschema.dll, defines a unique index on each of the tables using these three fields:

- Filename of the log file that was the source of this record (LOG_FILE)
- The line number of this record in that log file (LINE_NUM)
- The repetition number for this line, based on the syslog 'last message repeated n times' message (REPEAT_NUM)

This index prevents you from loading the same line number from the same named file more than once. This, combined with careful management of your log file names, should prevent duplication of log events in your database.

Adding other indexes to your database may enhance performance of your most common queries. Consult your database documentation for more information.

Table descriptions

This section maps firewall log messages to tables and columns and points to information you may wish to query for your reports. All messages that are mapped to a particular table are listed in the note at the end of the table. Messages that provide data for particular columns are listed in that column's description. The

tables contain messages for the IBM Firewall for AIX, the IBM Firewall for NT, and messages that are common to both firewalls.

For more information on firewall log messages, see “Appendix A. Messages” on page 101 .

In the Data Type column in the following descriptions, 'int' implies SMALLINT column type for DB2; 'long int' implies DB2 INTEGER type. A date-time Data Type implies DB2 TIMESTAMP. In the timestamp, the microseconds value will always be "000000".

If a description is marked *required*, a value must be specified to enter the record in the table.

The three columns that serve as the unique index and a column for receiving the log level indicator are omitted from these table descriptions because their definitions are identical and there is usually no reason to query them.

Table 1. ADMIN_ALERT. This table contains messages related to intrusion alerts from the a_alert.tbl file.

Column	Data Type	Short Description
DATE_TIME	date_time	Date and time for the action (required)
FIREWALL	char(100)	Fully qualified name of the firewall machine (required)
PID	int	AIX Process ID, NT thread ID (required)
MSG_NUM	int	Message number (required)
USERID	char(16)	User ID (ICA0001, ICA0002, ICA0003, ICA0004, ICA2001, ICA2002, ICA2003, ICA2026, ICA2043, ICA2068, ICA2167, ICA2168, ICA 2170, ICA2173, ICA3001, ICA3012, ICA3018)
ACTION	char(7)	connect (ICA3012) or bind (ICA3018)
NUM_COUNT	int	Number of authentication failures (ICA0001, ICA0002, ICA0003); number of log entries for TAG_MSG_NUM (ICA0004); number of days (ICA9000)
TAG_MSG_NUM	char (8)	Tag message number (ICA0004)
SRC_IP	char(15)	Source IP address (ICA2001, ICA2028, ICA2079, ICA2167, ICA3012, ICA3018)
DST_IP	char(15)	Destination IP address (ICA2028, ICA2079, ICA3012, ICA3018)
AUTH_METHOD	char(20)	Authentication Method (ICA2002, ICA2167, ICA2170)
NETWORK	char(25)	Network name (ICA2001, ICA2002, ICA2167)
HOST_NAME	char(100)	Host name (ICA0003, ICA2002)
TIMEOUT_SEC	int	Time-out seconds (ICA2026)
CONN_USERID	char(16)	Socks connect user name (ICA3001)
APPLICATION	char(30)	Application name such as telnet, ftp, ... (ICA2167, ICA2168, ICA2170, ICA3012)
Note: Related Messages: ICA0001 ICA0002 ICA0003 ICA0004 ICA0005 ICA0006 ICA0007 ICA0008 ICA0009 ICA0010 ICA0011 ICA0012 ICA0013 ICA0014 ICA0015 ICA0016 ICA0017 ICA0018 ICA0019 ICA0020 ICA0021 ICA0022 ICA1010 ICA2001 ICA2002 ICA2003 ICA2020 ICA2026 ICA2028 ICA2037 ICA2040 ICA2042 ICA2043 ICA2079 ICA2167 ICA2168 ICA2170 ICA2173 ICA3001 ICA3012 ICA3018 ICA9000 ICA9001		

Table 2. FILTER_ACTIVE_RULE. This table contains active FILTER rules from the f_rule.tbl file.

Column	Data Type	Short Description
DATE_TIME	date_time	Date and time for the action (required)
FIREWALL	char(100)	Fully qualified name of the firewall machine (required)
PID	int	AIX Process ID, NT thread ID (required)
MSG_NUM	int	Message number (required)
RULE_NUM	int	Rule number (required)
RULE	char(150)	Rule (required)
Note: Related Message: ICA1037		

Table 3. FILTER_INFO. This table contains error or general information messages related to FILTERS from the f_info.tbl file.

Column	Data Type	Short Description
DATE_TIME	date_time	Date and time for the action (required)
FIREWALL	char(100)	Fully qualified name of the firewall machine (required)
PID	int	AIX Process ID, NT thread ID (required)
MSG_NUM	int	Message number (required)
RULE_NUM	int	Filter rule number (ICA1005)
ERROR_NUM	int	System Error number -- AIX errno or Windows NT Last Error (ICA1007, ICA1008, ICA1009, ICA1011 ICA1013, ICA1015, ICA1021, ICA1023, ICA1024) Text corresponding to this error number is obtainable through the _strerror function. Text for Windows NT Last Error is available through the Format Message function or in Appendix A of the Win32 Programmer's Reference Volume 2.
LOAD_PATH	char(100)	Kernel extension load path (ICA1011, ICA1012)
DVC_DRV	char(25)	Device driver (ICA1021)
TERM_SIG	char(25)	Termination signal (ICA1260)
FILE_NAME	char(100)	File name (ICA1024)
RC	int	Internal firewall return code (ICA1019)
Note: Related Messages: ICA1001 ICA1002 ICA1003 ICA1005 ICA1007 ICA1008 ICA1009 ICA1011 ICA1012 ICA1013 ICA1014 ICA1015 ICA1016 ICA1017 ICA1019 ICA1021 ICA1022 ICA1023 ICA1024 ICA1200 ICA1260		

Table 4. FILTER_MATCH. This table contains the filter rules matched from the f_match.tbl file.

Column	Data Type	Short Description
DATE_TIME	date_time	Date and time for the action (required)
FIREWALL	char(100)	Fully qualified name of the firewall machine (required)
PID	int	AIX Process ID, NT thread ID (required)
MSG_NUM	int	Message number (required)
RULE_NUM	int	Rule number (required)
ACTION	char(6)	Rule type: permit, deny, etc.
DIRECTION	char(8)	Direction the packet was traveling inbound or outbound (required)
SRC_IP	char(15)	IP address of the sender(required)
DST_IP	char(15)	IP address of the recipient(required)

Table 4. FILTER_MATCH (continued). This table contains the filter rules matched from the f_match.tbl file.

Column	Data Type	Short Description
PROTOCOL	char(7)	High-level protocol such as UDP, IPIP, ICMP, TCP or TCP/ACK (required)
SRC_PORT	int	<ul style="list-style-type: none"> IP Packet type for ICMP Resource protocol port number for others (required)
DST_PORT	int	<ul style="list-style-type: none"> IP Packet code for ICMP Destination protocol port number for others (required)
ROUTING	char(5)	Routing affiliation of the packets: route or local (required)
INTERFACE	char(10)	Interface type: secure or nonsecure (required)
FRAGMENT	char(8)	Identifies if the packet is fragment or non-fragment (required)
TUNNEL_ID	int	Tunnel ID (required)
ENCRYPTION	char(7)	Encryption algorithm: DES_CBC or CDMF or none
BYTES	long int	Length of the specific packet (required)
Note: Related Message: ICA1036		

Table 5. FILTER_STATUS. This table contains information on status changes of filters from the f_stat.tbl file.

Column	Data Type	Short Description
DATE_TIME	date_time	Date and time for the action (required)
FIREWALL	char(100)	Fully qualified name of the firewall machine (required)
PID	int	AIX Process ID, NT thread ID (required)
MSG_NUM	int	Message number (required)
DAEMON	char(25)	AIX filter logging daemon (ICA1004), or Windows NT filter logging service.
VERSION	int	Version number (ICA1004, ICA1033)
RELEASE	int	Release number (ICA1004, ICA1033)
PACKET_LOGGING	char(8)	Status of packet logging enabled or disabled (ICA1035)
Note: Related Messages: ICA1004 ICA1032 ICA1033 ICA1034 ICA1035. The details of the filter rule updates (ICA1032) can be obtained from FILTER_ACTIVE_RULEtable.		

Table 6. NAT_INFO. This table contains Network Address Translation message information from the nat_info.tbl file.

Column	Data Type	Short Description
DATE_TIME	date_time	Date and time for the action (required)
FIREWALL	char(100)	Fully qualified name of the firewall machine (required)
PID	int	AIX Process ID, NT thread ID (required)
MSG_NUM	int	Message number (required)
VERSION	int	NAT Version number (ICA9033)
RELEASE	int	NAT Release number (ICA9033)
IP	char(15)	IP address (ICA9035, ICA9036)
Note: Related Messages: ICA9032, ICA9033, ICA9034, ICA9035, ICA9036		

Table 7. *PAGER_INFO*. This table contains information related to the paging feature of the Firewall, from the *pgr_info.tbl* file.

Column	Data Type	Short Description
DATE_TIME	date_time	Date and time for the action (required)
FIREWALL	char(100)	Fully qualified name of the firewall machine (required)
PID	int	AIX Process ID, NT thread ID (required)
MSG_NUM	int	Message number (required)
USERID	char(16)	User ID (ICA4036, ICA4174, ICA4175)
ERROR_NUM	int	System Error number - AIX errno or Windows NT Last Error (ICA4371)
PROGRAM	char(25)	Program name (ICA4000)
SIGNAL	int	Termination signal (ICA4000)
ID	int	Identifier (ICA4036)
PRIORITY	int	Priority (ICA4036)
PERIOD	int	Period (ICA4036)
RETRY_COUNT	int	Number of retries (ICA4036, ICA4313, ICA4314, ICA4364, ICA4365)
FROM_ENTRY	char(15)	Function name (ICA4036)
HOST_NAME	char(100)	Host name (ICA4174, ICA4175)
MESSAGE_TEXT	char(250)	Text of the page (ICA4036, ICA4353 - 4360, ICA4368, ICA4372)
SERVICE	char(25)	Service name (ICA4017)
SOCKET	int	Socket number (ICA4017)
FILENAME	char(100)	Filename (ICA4154, ICA4351, ICA4352)
Note: Related Messages: ICA4000 ICA4001 ICA4007 ICA4017 ICA4036 ICA4154 ICA4168 ICA4174 ICA4175, ICA4300 - 4303, ICA4305 - 4315, ICA4351 - 4360, ICA4362 - 4372)		

Table 8. *PROXY_FTP*. This table contains FTP action information from FTP sessions from the *p_ftp.tbl* file.

Column	Data Type	Short Description
DATE_TIME	date_time	Date and time for the action (required)
FIREWALL	char(100)	Fully qualified name of the firewall machine (required)
PID	int	AIX Process ID, NT thread ID (required)
MSG_NUM	int	Message number (required)
USERID	char(16)	User ID (required)
SRC_IP	char(15)	IP Address of the User (required)
DST_IP	char(15)	IP address of the remote machine (required)
ACTION	char(5)	File transfer action: put or get (required)
FILE_NAME	char(100)	File name
BYTES	long int	Amount of data transferred
SID	long int	Unique session ID (required)
Note: Related Message: ICA2075		

Table 9. *PROXY_HTTP*. This table contains HTTP action information from Proxy sessions from the *p_http.tbl* file.

Column	Data Type	Short Description
DATE_TIME	date_time	Date and time for the action (required)

Table 9. PROXY_HTTP (continued). This table contains HTTP action information from Proxy sessions from the p_http.tbl file.

Column	Data Type	Short Description
FIREWALL	char(100)	Fully qualified name of the firewall machine (required)
PID	int	AIX Process ID, NT thread ID (required)
MSG_NUM	int	Message number (required)
STATUS	int	Status (required)
SRC_IP	char(15)	IP Address of the user (required)
REQUEST	char(250)	Content of the HTTP request (required)
BYTES	long int	Amount of data transferred.

Note: Related Message: ICA2099

Table 10. PROXY_INFO. This table contains error or general information messages related to PROXY from the p_info.tbl file.

Column	Data Type	Short Description
DATE_TIME	date_time	Date and time for the action (required)
FIREWALL	char(100)	Fully qualified name of the firewall machine (required)
PID	int	AIX Process ID, NT thread ID (required)
MSG_NUM	int	Message number (required)
USERID	char(16)	User ID (ICA2018, ICA2019, ICA2057, ICA2058, ICA2166, ICA2177, ICA2172)
ERROR_NUM	int	System Error number - AIX errno or Windows NT Last Error (ICA2005, ICA2006, ICA2009, ICA2029, ICA2035, ICA2038, ICA2039, ICA2052, ICA2054, ICA2055, ICA2056, ICA2057, ICA2058, ICA2059, ICA2063, ICA2064, ICA2065, ICA2066, ICA2067, ICA2068, ICA2069, ICA2069, ICA2070, ICA2071, ICA2074, ICA2110, ICA2111, ICA2113, ICA2114, ICA2115, ICA2118, ICA2119, ICA2121, ICA2122, ICA2123, ICA2124, ICA2200, ICA2201, ICA2202, ICA2203) Text for errno (AIX System Errors) is obtainable via the _strerror function. Text for Windows NT Last Error is available through the Format Message function or in Appendix A of the Win32 Programmer's Reference Volume 2.
OPTION_VAL	char(20)	Option flag or parameter value (ICA2014, ICA2015, ICA2049, ICA2050)
TIME	char(15)	Invalid time interval (ICA2044, ICA2202)
RC	int	Internal Firewall return code (ICA2007, ICA2030, ICA2031, ICA2033, ICA2034, ICA2054, ICA2057, ICA2058, ICA2065, ICA2120 ICA2166, ICA2203)
INVOC_NAME	char(20)	Invocation name for socket or port at time system error occurred (ICA2055, ICA2056)
AUDIT_TYPE	char(7)	Unknown audit-type (7 hex digits) (ICA2004)
HOST_NAME	char(100)	Host name (ICA2106, ICA2107, ICA2126)
FILE_NAME	char(100)	File name (ICA2029, ICA2030, ICA2072, ICA2183, ICA2204, ICA2205, ICA2206, ICA2207)
LINE_NUM	int	Line number (ICA2029, ICA2030)

Table 10. PROXY_INFO (continued). This table contains error or general information messages related to PROXY from the p_info.tbl file.

Column	Data Type	Short Description
PROTOCOL	char(25)	Invalid protocol name (ICA2112, ICA2116)
CUSTOMIZED_ATTR	char(25)	Line number (ICA2105, ICA2106, ICA2125, ICA2166)
ODM_ERR_NUM	int	Error number from Object Data Manager (ICA2102, ICA2103, ICA2104, ICA2105, ICA2107, ICA2108, ICA2109, ICA2125)
APPLICATION (NT only)	char(30)	Application name (ICA2200, ICA2201, ICA2202, ICA2203, ICA2204, ICA2205, ICA2206, ICA2207)
CALLER (NT only)	char(25)	Calling function (ICA2200, ICA2201, ICA2202, ICA2203, ICA2204, ICA2205, ICA2206, ICA2207)
FAILED_IN (NT only)	char(25)	Failing function (ICA2201, ICA2203)
Note: Related Messages: ICA2004 ICA2005 ICA2006 ICA2007 ICA2009 ICA2014 ICA2015 ICA2018 ICA2019 ICA2023 ICA2029 ICA2030 ICA2031 ICA2032 ICA2033 ICA2034 ICA2035 ICA2038 ICA2039 ICA2044 ICA2045 ICA2046 ICA2047 ICA2048 ICA2049 ICA2050 ICA2051 ICA2052 ICA2053 ICA2054 ICA2055 ICA2056 ICA2057 ICA2058 ICA2059 ICA2060 ICA2061 ICA2062 ICA2063 ICA2064 ICA2065 ICA2066 ICA2067 ICA2068 ICA2069 ICA2070 ICA2071 ICA2072 ICA2073 ICA2074 ICA2100 ICA2102 ICA2103 ICA2104 ICA2105 ICA2109 ICA2110 ICA2111 ICA2112 ICA2113 ICA2114 ICA2115 ICA2116 ICA2117 ICA2118 ICA2119 ICA2120 ICA2121 ICA2122 ICA2123 ICA2124 ICA2125 ICA2126 ICA2127 ICA2166 ICA2171 ICA2172 ICA2183 ICA2200 ICA2201 ICA2202 ICA2203 ICA2204 ICA2205 ICA2206 ICA2207		

Table 11. PROXY_LOGIN. This table contains information (primarily regarding authentication) about successful PROXY logins from the p_login.tbl file.

Column	Data Type	Short Description
DATE_TIME	date_time	Date and time for the action (required)
FIREWALL	char(100)	Fully qualified name of the firewall machine (required)
PID	int	AIX Process ID, NT thread ID (required)
MSG_NUM	int	Message number (required)
USERID	char(16)	User ID (required)
APPLICATION	char(30)	Application name - telnet, ftp, ... (required)
AUTH_METHOD	char(15)	Authentication method (required)
NETWORK	char(25)	Network (secure/nonsecure - may have additional information also) (required)
HOST_NAME	char(100)	Host name (required)
Note: Related Messages: ICA2024 ICA2025 ICA2169		

Table 12. PROXY_STATUS. This table contains PROXY status information from the p_stat.tbl file.

Column	Data Type	Short Description
DATE_TIME	date_time	Date and time for the action (required)
FIREWALL	char(100)	Fully qualified name of the firewall machine (required)
PID	int	AIX Process ID, NT thread ID (required)
MSG_NUM	int	Message number (required)
USERID	char(16)	User ID (ICA2008, ICA2016, ICA2021)
SRC_IP	char(15)	Source IP address (ICA2000, ICA2008, ICA2010, ICA2011, ICA2012, ICA2013, ICA2141, ICA2180)
DST_IP	char(15)	Destination IP address (ICA2000, ICA2010, ICA2011, ICA2012, ICA2013)

Table 12. PROXY_STATUS (continued). This table contains PROXY status information from the p_stat.tbl file.

Column	Data Type	Short Description
REMOTE_HOST	char(100)	Remote host name (from perspective of firewall machine) (ICA2021, ICA2022, ICA2027)
SID (NT only)	int	Session identifier (ICA2177, ICA2180, ICA2181 ICA2182)
SOCKET (NT only)	char(25)	Socket name (ICA2177)
RC (NT only)	int	Return or reason code (ICA2181, ICA2182)
CMD (NT only)	char(36)	SMTP card (ICA2182)
Note: Related Messages: ICA2000 ICA2010 ICA2011 ICA2012 ICA2013 ICA2016 ICA2021 ICA2022 ICA2027 ICA2097 ICA2098 ICA2141 ICA2163 ICA2164 ICA2165 ICA2177 ICA2180 ICA2181 ICA2182		

Table 13. SERVER_INFO. This table contains information about Configuration Server status and activities from the srv_info.tbl file.

Column	Data Type	Short Description
DATE_TIME	date_time	Date and time for the action (required)
FIREWALL	char(100)	Fully qualified name of the firewall machine (required)
PID	int	AIX Process ID, NT thread ID (required)
MSG_NUM	int	Message number (required)
USERID	char(16)	User ID (ICA9003, ICA9004)
ERROR_NUM	int	System Error number – AIX errno or Windows NT Last Error (ICA9008, ICA9009) Text for errno (AIX System Errors) is obtainable with the strerror function. Text for Windows NT Last Error is available through the Format Message function or in Appendix A of the Win32 Programmer's Reference Volume 2.
Note: Related Messages: ICA9003 ICA9004 ICA9005 ICA9006 ICA9007 ICA9008 ICA9009 ICA9010 ICA9011 ICA9012 ICA9013 ICA9014 ICA9015		

Table 14. SESSION. This table contains SOCKS and PROXY session start/stop information from the session.tbl file.

Column	Data Type (length)	Short Description
DATE_TIME	date_time	Date and time for the action (required)
FIREWALL	char(100)	Fully qualified name of the firewall machine (required)
PID	int	AIX AIX Process ID, NT thread ID, NT thread ID (required)
MSG_NUM	int	Message number (required)
USERID	char(16)	User ID (required)
SERVICE_TYPE	char(10)	Service type: socks or proxy (required)
APPLICATION	char(30)	Application name - telnet, ftp, (required)
SRC_IP	char(15)	IP Address of the user (required)
DST_IP	char(15)	IP address of the remote machine (required)
SESSION_EVENT	char(5)	<ul style="list-style-type: none"> begin when a session is established. end when a session is terminated. (required)
BYTES	long int	Amount of data transferred during the session. If the application is telnet, this will be 0.

Table 14. SESSION (continued). This table contains SOCKS and PROXY session start/stop information from the session.tbl file.

Column	Data Type (length)	Short Description
SID	long int	Unique session identifier, generated by the Firewall, based on clock time.
<p>Note:</p> <p>Related Messages:</p> <ul style="list-style-type: none"> • Safemail Session Start: ICA2178 • Safemail Session Stop: ICA2179 • Socks Session Start: ICA3011 • Socks Session Stop: ICA3015 • Proxy Telnet Session Start: ICA2036 (AIX Logs) ICA2208, ICA2218 (NT Logs) • Proxy Telnet Session Stop: ICA2077 (AIX Logs) ICA2209, ICA2219 (NT Logs) • Proxy FTP Session Start: ICA2041 (AIX Logs) ICA2208, ICA2218 (NT Logs) • Proxy FTP Session Stop: ICA2076 (AIX and NT Logs) <p>Details of Socks FTP session actions are in SOCKS_FTP table. Details of Proxy FTP session actions are in PROXY_FTP.</p>		

Table 15. SOCKS_FTP. This table contains SOCKS FTP action information from FTP sessions from the s_ftp.tbl file.

Column	Data Type	Short Description
DATE_TIME	date_time	Date and time for the action (required)
FIREWALL	char(100)	Fully qualified name of the firewall machine (required)
PID	int	AIX AIX Process ID, NT thread ID (required)
MSG_NUM	int	Message number (required)
USERID	char(16)	User ID (required)
SRC_IP	char(15)	IP Address of the User (required)
DST_IP	char(15)	IP address of the remote machine (required)
DATA_BIND	char(5)	<ul style="list-style-type: none"> • 'start' when data bind is established.(ICA3010) • 'stop' when data bind is terminated.(ICA3014) (required)
BYTES	long int	Amount of data transferred.
<p>Note: Related Messages: ICA3010 ICA3014</p>		

Table 16. SOCKS_INFO. This table contains error or general information messages related to Socks from the s_info.tbl file.

Column	Data Type	Short Description
DATE_TIME	date_time	Date and time for the action (required)
FIREWALL	char(100)	Fully qualified name of the firewall machine (required)
PID	int	AIX Process ID, NT thread ID (required)
MSG_NUM	int	Message number (required)
USERID	char(16)	User ID (ICA3044, ICA3045, ICA3046, ICA3047, ICA3049)
ACTION	char(7)	connect (ICA3044, ICA3049) or bind (ICA3046, ICA3047)

Table 16. SOCKS_INFO (continued). This table contains error or general information messages related to Socks from the s_info.tbl file.

Column	Data Type	Short Description
ERROR_NUM	int	System Error number - AIX errno (ICA3013, ICA3019, ICA3031, ICA3032, ICA3040, ICA3044, ICA3101, ICA3102, ICA3103, ICA3104, ICA3106, ICA3107, ICA3108, ICA3122, ICA3124, ICA3125, ICA3126, ICA3128)
SRC_HOST	char(25)	Source host name (ICA3019, ICA3035)
DST_HOST	char(25)	Destination host name (ICA3016, ICA3045)
SRC_IP	char(15)	Source address (ICA3042, ICA3043, ICA3044, ICA3045, ICA3046, ICA3047, ICA3049)
DST_IP	char(15)	Destination address (ICA3044, ICA3045, ICA3046, ICA3047, ICA3049)
LINE_NUM	int	Line number (ICA3022, ICA3023, ICA3024, ICA3025, ICA3026, ICA3109, ICA3110, ICA3111, ICA3112, ICA3115, ICA3116, ICA3117, ICA3118, ICA3119, ICA3120); or Number of lines (ICA3113)
EXEC_STATUS	int	Exec status (ICA3027)
CMD	char(36)	Command, such as login (ICA3027, ICA3039, ICA3042, ICA3044, ICA3048) note: for ICA3042, the command is in hexadecimal format
FILE_NAME	char(100)	File name (ICA3030, ICA3032, ICA3105, ICA3109, ICA3110, ICA3111, ICA3112, ICA3113, ICA3114, ICA3115, ICA3116, ICA3117, ICA3118, ICA3119, ICA3120)
APPLICATION	char(30)	Application name - telnet, ftp... (ICA3044, ICA3045, ICA3049)
VERSION	char(10)	Socks version number in hex (ICA3043)
<p>Note: Related Messages: ICA3013 ICA3016 ICA3017 ICA3019 ICA3022 ICA3023 ICA3024 ICA3025 ICA3026 ICA3027 ICA3030 ICA3031 ICA3032 ICA3033 ICA3035 ICA3039 ICA3040 ICA3041 ICA3042 ICA3043 ICA3044 ICA3045 ICA3046 ICA3047 ICA3048 ICA3049 ICA3052 ICA3101 ICA3102 ICA3103 ICA3104 ICA3105 ICA3106 ICA3107 ICA3108 ICA3109 ICA3110 ICA3111 ICA3112 ICA3113 ICA3114 ICA3115 ICA3116 ICA3117 ICA3118 ICA3119 ICA3120 ICA3121 ICA3122 ICA3123 ICA3124 ICA3125 ICA3126 ICA3127 ICA3128</p>		

Table 17. SSL_INFO. This table contains information about SSL status and activities from the ssl_info.tbl file.

Column	Data Type	Short Description
DATE_TIME	date_time	Date and time for the action (required)
FIREWALL	char(100)	Fully qualified name of the firewall machine (required)
PID	int	AIX Process ID, NT thread ID (required)
MSG_NUM	int	Message number (required)
Client_IP	char(15)	IP address of the client
<p>Note: Related Messages: ICA5015 ICA5022 ICA5023 ICA5028 ICA5029 ICA5036 ICA5039 ICA5060 ICA5063 ICA5082 ICA5120</p>		

Table 18. SU. This table contains details about SU activities from the su.tbl file if you are loading an AIX su log.

Column	Data Type	Short Description
DATE_TIME	date_time	Date and time for the action (required) Because AIX does not record the year in the su log file, the year portion of the DATE_TIME column is set to either the current year or the previous year, based on the month/day settings (if month/day is later than current month/day, assume it is last year.)
FROM_USERID	char(16)	User ID (required)
TO_USERID	char(16)	User ID (required)
LOGIN_STATUS	char(7)	Status of login attempt: success or failure (required)

Table 19. TUNNEL_CONTEXT. This table contains active TUNNEL context specifications from the t_cntxt.tbl file.

Column	Data Type	Short Description
DATE_TIME	date_time	Date and time for the action (required)
FIREWALL	char(100)	Fully qualified name of the firewall machine (required)
PID	int	AIX Process ID, NT thread ID (required)
MSG_NUM	int	Message number (required)
TUNNEL_ID	long int	Tunnel ID (required)
SRC_IP	char(15)	Source IP address (required)
DST_IP	char(15)	Destination IP address (required)
ENCRYPTION	char(7)	Encryption algorithm DES_CBC or CDMF
Note: Related Message: ICA1043		

Table 20. TUNNEL_POLICY. This table contains TUNNEL policy statements from the t_policy.tbl file.

Column	Data Type	Short Description
DATE_TIME	date_time	Date and time for the action (required)
FIREWALL	char(100)	Fully qualified name of the firewall machine (required)
PID	int	AIX Process ID, NT thread ID (required)
MSG_NUM	int	Message number (required)
POLICY	char(60)	Policy statement read from fwpolicy file (required)
Note: Related Message: ICA1040		

Table 21. TUNNEL_STATUS. This table contains information on status changes of TUNNELS from the t_stat.tbl file.

Column	Data Type	Short Description
DATE_TIME	date_time	Date and time for the action (required)
FIREWALL	char(100)	Fully qualified name of the firewall machine (required)
PID	int	AIX Process ID, NT thread ID (required)
MSG_NUM	int	Message number (required)
SESSION_SCKT	long int	Session socket port (for ICA1038)
MASTER_SCKT	long int	Master socket port (for ICA1038)
TUNNEL_ID	long int	Tunnel ID deleted (for ICA1041)

Table 21. TUNNEL_STATUS (continued). This table contains information on status changes of TUNNELS from the t_stat.tbl file.

Column	Data Type	Short Description
Note:		
Related Messages: ICA1038 ICA1039 ICA1041 ICA1042		
<ul style="list-style-type: none">• The details of the policy defined(ICA1039) can be obtained from TUNNEL_POLICY table.• The details of the tunnel context defined(ICA1042) can be obtained from TUNNEL_CONTEXT table.		

Chapter 3. Providing Your Own Authentication Methods

This chapter gives you information on providing your own authentication methods.

There are two methods for user-supplied authentication:

1. Using executables **fwuserpt** and **fwuserau**, which authenticates a user based on a response to a single prompt.
2. Using the functions **fwprompt**, **fw_tn_authenticate**, and **fw_ftp_authenticate**, which authenticates a user based on responses to one or more prompts. This method is known as user-supplied iteration because the prompting is an iterative process.

You can use either method but not both. If **fwuserpt** and **fwuserau** are present in `/usr/bin`, then method 1 is used.

User-Supplied Authentication

To use user-supplied authentication as an authentication method, the firewall administrator must provide two executables: **fwuserpt** and **fwuserau**. The **fwuserpt** code provides the text that will prompt the user for an authentication token. The **fwuserau** code authenticates the user based on the response to the prompt.

If you choose the user-supplied authentication method for a firewall proxy user, the IBM Firewall takes these actions when that user logs on :

- Prompts the user for a user name.
- Invokes **fwuserpt** passing the user name as the input parameter.
- **fwuserpt** executes a `printf` statement to display a prompt to the user.
- Receives the `printf` string and displays it on the user console.
- Reads the user response, which may be multiple tokens on a single line.
- Invokes **fwuserau** passing the User Name and all the tokens read from the user terminal.
- Accepts or rejects the user based on the return code from **fwuserau**.

FWUSERPT and FWUSERAU Specifications

The executables, **fwuserpt** and **fwuserau** reside in `/usr/bin`. These subroutines are supported when compiled in an AIX machine compatible with the version of the firewall the subroutines are intended to run on. Once executed, the ownership is transferred to root.

fwuserpt takes user name as the input. It performs a database lookup or calculation and outputs a string using a `printf` statement. For example, if John is the user, **fwuserpt** can create one of the following as output:

- Please enter your secret code for authentication
- Secret code is required for John
- Secret code corresponding to 1345 is required for John. (1345 is a string associated with John.)

The return codes for **fwuserpt** and **fwuserau** are zero if successful and non-zero if unsuccessful.

The input to **fwuserau** are the user name and the strings of 'password' supplied by the user. If the password consists of a sequence of strings as in the case of Secure Key, they are in `argv[2]`, `argv[3]`, `argv[4]`, `argv[5]` and so on.

The string **fwuserpt** issues with `printf` must not contain any special character like `\n` or `\r`, otherwise, the result is unpredictable. It must contain a `fflush` statement after `printf`. The **fwuserau** must not contain any print statements, otherwise, the result is unpredictable.

Example of fwuserpt and fwuserau

The following is an example of **fwuserpt** and **fwuserau** with authentication performed.

Compile the following and name the output file **fwuserpt**.

```
int main (int argc, char **argv)
{char *user = NULL; /* name of user to be authenticated */
  user = argv [1];
  if (user == NULL)
  {return 1;}
  /* Note, if you cannot validate the userid, return
  1 and fwuserau will not be called */
  printf ("User Supplied auth invoked. Please supply your password.");
  (void) fflush(stdout);
  return 0;}
```

Compile the following and name the output file **fwuserau**.

```
int main(int argc, char **argv)
{char *user = NULL; /* name of user to be authenticated*/
  user = argv[1];
  if (user == NULL)
  {return 1;}
  /* retrieve the authentication token from
  argv[2], argv[3], etc depending on the
  interface, ie, the number of tokens the user is expected to
  input at the prompt and validate the user.
  return 0 if successful.
  return 1 if unsuccessful*/}
```

Secure Key as an Example of User-Supplied Authentication

Sample code using Secure Key as an example of user-supplied authentication is provided in the `/usr/lpp/FW/sample` directory. These files are:

- `makefile.ex`
- `fwuserpt.c`
- `fwuserau.c`

These files have been used to test our user-supplied authentication API code and are provided as is.

User-Supplied Iteration Support

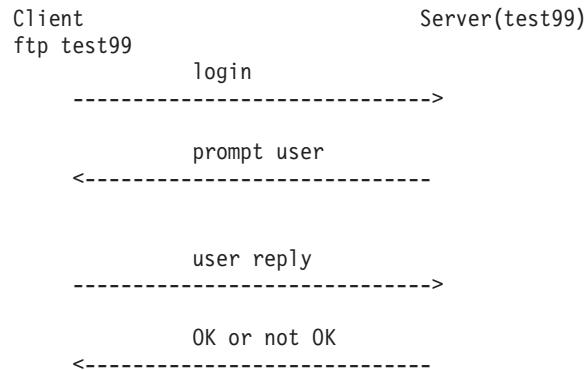
User-supplied iteration enables you to require multiple responses from a user attempting to logon, instead of just one prompt and one reply.

This user-supplied authentication method provides support for iterating through a loop during authentication for telneting. So, for example you can have telnet do a looping instead of doing just one prompt and one reply from users.

The following diagram depicts the user-supplied support for telnet :



The following diagram depicts the user-supplied support for ftp :



The following sections explain how to accomplish iterative prompting.

Library Requirements

You must supply the library functions that the firewall invokes. The name of the shared library file is **fwuser.o**. This file must reside in `/usr/lib`. In addition, **fwuser.exp** must be present in `/usr/lib`. This library must contain the following functions:

- fw_prompt
- fw_tn_authenticate

- fw_ftp_authenticate

Note: Iteration does not apply to FTP; however, FTP authentication is supported.

During installation of the IBM Firewall, copies of **fwuser.o** and **fwuser.exp** are installed in /usr/lib. If a copy of **fwuser.o** already exists in /usr/lib, it will not be replaced during installation.

When you invoke the IBM-supplied **fwuser.o**, a message reminding you to supply your own version of user-written authentication is put into the local4 log.

The IBM- supplied **fwuser.o** denies authentication for FTP and telnet.

Details of fw_prompt

fw_prompt authenticates the user using FTP.

fw_prompt prompts the user for the returned string, *password*, for example.

fw_prompt verifies that a name is authorized in the database and displays messages to the user.

fw_prompt takes two arguments, a pointer to username (characters) and ret_code. ret_code is a pointer to a data structure called **fw_ret_struct** which is defined in **fwuser.h**. **fwuser.h** can be found in the /usr/lpp/FW/sample subdirectory.

In the argument ret_code, the req_rsp_code is set to FW_AUTH_REQ (request for prompt for username).

- **fw_prompt** uses the username to compute a prompt in the form of a string and puts it in ret_code->return_str to be displayed to the user.
- ret_code->return_str must not contain any special character like '\n'.
- If verification of username is successful, **fw_prompt** puts a string in ret_code->return_str to be displayed to user and sets the req_rsp_code to be FW_AUTH_OK. Setting req_rsp_code to anything else means the authentication failed.

Following is an example of the function for fw_prompt. This can be found in /usr/lpp/FW/sample/fwprompt.c.

```
#include <stdio.h>
#include <stdlib.h>
#include "fwuser.h"

int fw_prompt (char *username, struct fw_ret_struct *ret_code)
{
    strcpy(ret_code->return_str, "Please enter password");
    ret_code->req_rsp_code = FW_AUTH_OK;
    return FW_AUTH_OK;
}
```

Details of fw_tn_authenticate

fw_tn_authenticate authenticates the user using telnet.

fw_ftp_authenticate is the function that authenticates a user using FTP.

fw_tn_authenticate takes three arguments:

- a pointer to username
- a pointer to response
- a pointer to a data structure called `ret_code`.

ret_code, also called **fw_ret_struct**, is defined in **fwuser.h**. The description of the various arguments are:

- **username** always points to a NULL terminated string or user ID of the user.
- **response** points to a NULL terminated string or NULL.

If **response** points to NULL and the `req_rsp_code` is set to `FW_AUTH_REQ`, this is the first time **fw_tn_authenticate** is called for the user specified by the username. For example, when a telnet session is initiated, before any prompt is displayed, **fw_tn_authenticate** is called with `response` set to NULL and `req_rsp_code` set to `FW_AUTH_REQ`,

The `ret_code` is used to pass information back and forth between the firewall and `fw_tn_authenticate`.

`Ret_code` can have these values:

FW_AUTH_REQ

The initial setting; indicates the first call to `fw_tn_authenticate`

FW_AUTH_OK

The user has been verified; authentication is successful. `fw_tn_authenticate` is not called again.

FW_AUTH_FAILED

The user has not been verified. `fw_tn_authenticate` is not called again for that user. The user is rejected.

FW_AUTH_MISSING

`fwuser` is missing and authentication will fail.

If `ret_code` is set to anything other than these values, `fw_tn_authenticate` is called again. You should define a code, such as `FW_AUTH_INIT`, to indicate authentication is continuing and `fw_tn_authenticate` should be called again.

The **return_str** is the string firewall will display to the user for response. This **return_str** must NOT contain any special characters like `'\n'`.

For example, if an authentication uses a sequence of passwords defined by the user, the user can define `FW_AUTH_CONT_REQ` as 3 and `FW_AUTH_INIT_REQ` as 2 and put these in **fwuser.h**. When **fw_tn_authenticate** is first called, the second parameter is set to NULL and `req_rsp_code` is set to `FW_AUTH_REQ`. Then **fw_tn_authenticate** can put a string like "Enter the initial code" in the `return_str` and set `req_rsp_code` to be `FW_AUTH_INIT_REQ`.

When **fw_tn_authenticate** is called again, the second parameter will point to a string of response and the `req_rsp_code` will be `FW_AUTH_INIT_REQ`. If further input from the user is needed, **fw_tn_authenticate** can put "enter your second response" and set the `req_rsp_code` to `FW_AUTH_CONT_REQ`. When **fw_tn_authenticate** is called again, the second parameter will point to the second response given by the user and the `req_rsp_code` will be `FW_AUTH_CONT_REQ`.

If **fw_tn_authenticate** is satisfied with the response, **fw_tn_authenticate** will set req_rsp_code to FW_AUTH_OK and return FW_AUTH_OK.

If **fw_tn_authenticate** is not satisfied with the user response, it will set req_rsp_code to be FW_AUTH_FAILED and return.

Here is an example of **fw_tn_authenticate** that implements the above scenario. In this example, the user is first asked to enter "password" and the second authentication asks the user to enter the changed "password".

```
/* This is an example of two iteration authentications. It first asks user to input
a response and based on the response, asks the user for a second response
for authentication */
#include <stdio.h>
#include <stdlib.h>
#include "fwuser.h"
int fw_tn_authenticate (char *username, char *response, struct fw_ret_struct *ret_code)
{
    if (username == NULL) {
        return FW_AUTH_FAILED;
    }
    if (ret_code == NULL) {
        return FW_AUTH_FAILED;
    }
    memset(ret_code->return_str, 0x00, sizeof(ret_code->return_str));
    if ((response == NULL) && (ret_code->req_rsp_code == FW_AUTH_REQ)) {
        ret_code->req_rsp_code = FW_AUTH_INIT_REQ;
        strcpy(ret_code->return_str, "Please enter password");
        return FW_AUTH_OK;
    }
    else {
        switch (ret_code->req_rsp_code) {
            case FW_AUTH_INIT_REQ:
                if (strcmp(response, "password") == 0) {
                    ret_code->req_rsp_code = FW_AUTH_CONT_REQ;
                    strcpy(ret_code->return_str, "Please enter password");
                    return FW_AUTH_OK;
                } else {
                    ret_code->req_rsp_code = FW_AUTH_FAILED;
                    return FW_AUTH_FAILED;
                } /* endif */
                break;
            /* put other case statement defined in fwuser.h */
            case FW_AUTH_CONT_REQ:
                if (strcmp(response, "password") == 0) {
                    ret_code->req_rsp_code = FW_AUTH_OK;
                }
                else {
                    ret_code->req_rsp_code = FW_AUTH_FAILED;
                }
                return FW_AUTH_OK;
            /* put other case statement defined in fwuser.h */
            default:
                ret_code->req_rsp_code = FW_AUTH_FAILED;
                return FW_AUTH_FAILED;
                break;
        } /* switch */
    }
    return FW_AUTH_FAILED;
}
```

Details of fw_ftp_authenticate

fw_ftp_authenticate authenticates the user using FTP.

The argument taken by **fw_ftp_authenticate** is identical to that of **fw_tn_authenticate**. It can only return FW_AUTH_FAILED or FW_AUTH_OK. Any value other than FW_AUTH_OK in req_rsp_code will fail.

If the authentication is successful, the value in req_rsp_code must be set to FW_AUTH_OK and the returned value of the function is FW_AUTH_OK. Returning non-zero by the function or setting req_rsp_code to be anything other than FW_AUTH_OK means authentication failed. This file can be found in /usr/lpp/FW/sample/fwauthen.c.

Following is an example of **fw_ftp_authenticate**.

```

/*
 * The following is an example of user authentication. It uses a
 * two stage authentication method. This procedure is provided as
 * is. The first time this procedure is invoked, it asks the user to respond with "password".
 * If the user responds properly, it asks the user
 * to respond with "changed password". If the user responds properly, then
 * the user is authenticated. Otherwise, the authentication fails.
 * FW_AUTH_INIT_REQ and FW_AUTH_CONT_REQ are user defined constants that
 * are defined in fwuser.h. The IBM Firewall does not use
 * these two constants. The constants are defined in fwuser.h.
 * The IBM Firewall uses (and user must not redefine) FW_AUTH_FAILED, FW_AUTH_OK and FW_AUTH_REQ
 * The fwuser.o that is being installed was not compiled using this program.
 */

#include <stdio.h>
#include <stdlib.h>
#include "fwuser.h"

int fw_tn_authenticate (char *username, char *response, struct fw_ret_struct *ret_code)
{
    if (username == NULL) {
        return FW_AUTH_FAILED;
    }
    if (ret_code == NULL) {
        return FW_AUTH_FAILED;
    }

    memset(ret_code->return_str, 0x00, sizeof(ret_code->return_str));
    if ((response == NULL) && (ret_code->req_rsp_code == FW_AUTH_REQ)) {
        ret_code->req_rsp_code = FW_AUTH_INIT_REQ;
        /*
         In here, the program makes a computation or database lookup
         for username. It then comes up with a prompt for the user to
         enter the response. In this example, the user is asked to
         enter 'password' as a string. It can be changed to
         'please enter your password' or 'please enter your code'
         or any appropriate message to prompt the user for response.
        */
        strcpy(ret_code->return_str, "Please enter password");
        return FW_AUTH_OK;
    }
    else {
        switch (ret_code->req_rsp_code) {
            case FW_AUTH_INIT_REQ:/*
                The program is checking the response to see if it is valid.*/
                if (strcmp(response, "password") == 0) {
                    ret_code->req_rsp_code = FW_AUTH_CONT_REQ;/*
                        In this example, the first response from the user is valid and the user
                        is asked to enter the 'changed password'. If the administrator,
                        after looking up the user's credential, determines that the
                        password has expired, a prompt requesting user change the
                        password can be issued.
                    */

```

```

    */
    strcpy(ret_code->return_str, "Please enter changed password");
    return FW_AUTH_OK;
} else {
    ret_code->req_rsp_code = FW_AUTH_FAILED;
    return FW_AUTH_FAILED;
}
break;
/* put other case statement defined in fwuser.h */
case FW_AUTH_CONT_REQ:
    /*
    Computation is done to check the validity of
    the response.*/
    if (strcmp(response, "changed password") == 0) {
        ret_code->req_rsp_code = FW_AUTH_OK;
        return FW_AUTH_OK;
    }
    else {
        ret_code->req_rsp_code = FW_AUTH_FAILED;
        return FW_AUTH_FAILED;
    }
    /* put other case statement defined in fwuser.h */
default:
    ret_code->req_rsp_code = FW_AUTH_FAILED;
    return FW_AUTH_FAILED;
    break;
} /* switch */
}
return FW_AUTH_FAILED;
}

/* The following procedure is called after user responses to fwprompt. */
/* It only check to see if the response is password */

int fw_ftp_authenticate (char *username, char *response, struct fw_ret_struct *ret_code)
{
    if (username == NULL) return FW_AUTH_FAILED;
    if (response == NULL) return FW_AUTH_FAILED;

    /* checking the validity of the response based on the return */
    if (strcmp(response, "password") == 0) {
        ret_code->req_rsp_code = FW_AUTH_OK;
        return FW_AUTH_OK;
    }
    else {
        ret_code->req_rsp_code = FW_AUTH_FAILED;
        return FW_AUTH_FAILED;
    }
}
}
}

```

Sample Makefile for fwuser.o

Following is an example of a makefile for making **fwuser.o**. In this example, **fwauthen.c** contains **fw_tn_authenticate** and **fw_ftp_authenticate**. **fwprompt.c** contains **fw_prompt**. Call this makefile **Makefile.lib**. This information can be found in **/usr/lpp/FW/sample/Makefile.lib**.

```

CDEBUGFLAGS=

LDFLAGS=
CDEBUGFLAGS=

HASSTDLIB=-DHASSTDLIB

```

```

LIB=fwuser.o
LIBOBS=fwauthen.o fwprompt.o

CFLAGS=$(CDEBUGFLAGS) $(HASSTDLIB)

all: $(LIB)

$(LIB): $(LIBOBS)
    cc $(HASSTDLIB) -o fwuser.o $(LIBOBS) -bE:fwuser.exp -bM:SRE -e _t

fwauthen.o: fwauthen.c fwuser.h
fwprompt.o: fwprompt.c fwuser.h

```

fwuser.o can be obtained by executing the following command after renaming `fwuser.exp.df` to `fwuser.exp`:

- `make -f Makefile.lib`

The sample files `Makefile.lib`, `fwuser.h`, `fwauthen.c` and `fwprompt.c` are in `/usr/lpp/FW/sample`. A copy of `fwuser.exp.df` is also in `/usr/lpp/FW/sample`.

```

]> --> /* -- BEGIN COPYRIGHT -- * * *
IBM Firewall for AIX - Version 3 Release 1.0 * *
5765-C16
(C) Copyright IBM Corp. 1994, 1997 *
All Rights Reserved *
Licensed Material - Property of IBM * *
US Government Users Restricted Rights - *
Use, duplication or disclosure restricted by GSA ADP
Schedule Contract *
with IBM Corp. * *
-- END COPYRIGHT -- */

/* * The following is the header file needed to compile
fwauthen.c and * fwprompt.c.
This procedure is provided as is. *
Users must not change the definition of FW_AUTH_FAILED,
FW_AUTH_OK and *
FW_AUTH_REQ.
However, users can add some other definitions which are *
relevant to the user's authentication method.
FW_AUTH_INIT_REQ and *
FW_AUTH_CONT_REQ are added and are referenced by fwauthen.c
in the *
sample given.
*/#ifndef _H_FWUSER #define _H_FWUSER_
    #define FW_AUTH_OK 0
/* Authentication successful, */
/* put in ret_req_code */
/* by fw_tn_authenticate. */
/* Do NOT redefine this value. */
    #define FW_AUTH_REQ 1
/*Requests authentication. */
/* Do NOT redefine this value. */
    #define FW_AUTH_INIT_REQ 2
/* Used by fw_tn_authenticate. */
/* Can be redefined by user. */
    #define FW_AUTH_CONT_REQ 3
/* Used by fw_tn_authenticate. */
/* Can be redefined by user. */
    #define FW_NO_DISP_RSP 0x01
/* If opt for no Response: */
/* set this in sec_rc */

```

```

#define FW_AUTH_MISSING 98
/* Set if fwuser.o was not modified */
/* for user supplied authentication. */
/* Do NOT redefine this value. */
#define FW_AUTH_FAILED 99
/* Authentication failed. */
/* Do NOT redefine this value. */
Examples of additional return codes to be defined for use with
fw_tn_authenticate.
#define FW_AUTH_UPDATE 4
#define FW_AUTH_CHANGE 5
/* struct fw_ret_struct { int req_rsp_code;
/* Request / response code */
int sec_rc;
/* Reserved for future use */
char return_str[254];
/* return string to be displayed for */
/* user. */ };
#endif /* _H_FWUSER_ */

End of file

```

To compile the sample files, rename fwuser.exp.df to fwuser.exp.

Migration

If both of the files /usr/bin/fwuserau and /usr/bin/fwuserpt are present, the functions in /usr/lib/fwuser.o will not be called. If you want to use the functions in fwuser.o, fwuserau or fwuserpt must be removed or renamed.

Chapter 4. Using Network Management with the IBM Firewall

This chapter describes how to use the Simple Network Management Protocol (SNMP) to monitor your IBM Firewall resources.

SNMP is an open application protocol used in a TCP/IP environment for managing network functions. This chapter assumes familiarity with SNMP. The IBM Firewall uses the Management Information Base (MIB) and the SNMP Subagent to monitor the status of servers (INETD, FWPAGERD, FWMAILD, NAMED, PHTTDP, and SOCKD) and critical log records.

Management information is the abstraction of managed resources. These resources are defined as managed objects. The collection of managed objects is called a MIB. The MIB acts as the information store of the definition and specification of SNMP managed objects. The SNMP Subagent is a program or process that handles a specific portion of the MIB. The MIB registers with the SNMP agent so the agent knows where to send requests for the variables requested.

The SNMP agent performs all management operations as inspections or alterations of managed objects. The management operations are **get** or **getnext**. However, the firewall subagent does not support **set**.

The subagent can also emit unsolicited messages through "traps".

To set up SNMP traps:

1. Edit the `/etc/snmpd.conf` file

There is a line in this file that defines where the traps are sent. It looks like this:

```
trap public 1.23.456.78 1.2.3 fe
```

where

```
community name is public
IP address of the manager to receive the trap is 1.23.456.78
view=1.2.3
traps to be blocked
```

The last field (fe) indicates what traps you want to block:

```
fe block no traps (1111 1110)
7e block coldStart trap (0111 1110)
be block warmStart trap (1011 1110)
3e block coldStart trap and warmStart trap (0011 1110)
```

There are many coldStart traps that are issued when SNMP starts. The mask of 7e may be used to block the coldStart traps.

2. Modify this trap line to point to an SNMP Manager address as follows:

```
trap public 9.67.128.41 1.2.3 fe
```

3. The SNMP manager administrator sets up an SNMPTRAP listener application on the machine at the address specified on the trap statement in the `/etc/snmpd.conf` file.
4. When any of the six monitored servers changes states from Running to Not Running or vice versa, a trap is sent to the address defined in the trap statement in the `/etc/snmpd.conf` file on the firewall machine.

Note: Only one trap statement is allowed. If more than one trap statement is added, there is no error message on startup, but only the first trap definition entry specifications are used.

These are servers and codes received when a trap is received on the manager and monitored by the firewall SUBAGENT. (For example, 69 6E 65 74 64 is not running.) The numerical codes are the ASCII representation of the server names.

Not Running	6E 6F 74 20 52 75 6E 6E 49 62 67		
Running	6E 49 6E 47		
		Not Running	
INETD	69 6E 65 74 64	/	/
FWPAGERD	66 77 70 61 67 65 72 64	/	/
FWMAILD	66 77 6D 61 69 6C 64	/	/
NAMED	6E 61 6D 61 64	/	/
PHTTPD	70 68 74 74 70 64	/	/
SOCKD	73 6F 63 66 64	/	/

The subagent monitors the firewall log facility for -e, -i, and -w alert messages of emergency warning or information levels, and notifies the SNMP daemon of these alerts. A firewall log facility of at least information priority should be created.

SNMP trappable events

```
ICA0001e Threshold conditions for authentication failures
         have been satisfied
ICA0002e Threshold conditions for detecting a specific log
         message have been satisfied
ICA0003e Threshold conditions for authentication failures
         from any specific host have been satisfied
ICA0004e Threshold conditions for detecting a specific log
         message have been satisfied
ICA0012e Daemon is abending or received terminate signal.
         Previous log messages would provide detail.
ICA1010i The /usr/sbin/fwlogd daemon must be started under
         root authority
ICA2001e A user, without an account, attempted to use ftp
         proxy from the network
ICA2002e Firewall is unable to authenticate the indicated
         username using the specified authentication method
ICA2026i Connection attempt timed out for specified user.
         Potential network routing problem or remote host
         is not available
ICA2043i Authentication type for this user is 'password'
         and no password was found.
ICA3001e Real user is ident username,not socks connect username
ICA3012w refused -- Connect from ser(real_user)@src_addr
         to dst_addr (application)
ICA9000i Internet Connection IBM Firewall (FW) evaluation
         expires in number of days
```

SystemView Agent and SystemView Mapper are installed before the SNMP subagent can be invoked.

Note: The hostname must be known to itself. The /etc/hosts should have an entry of itself.

An SNMP manager is refreshed when an SNMP manager is added or deleted from the IBM Firewall.

If the filter is active, start an SNMP manager by:

1. Creating an object of the network manager with IP address.

2. Assuming there is an object for this firewall.
3. Creating a service to permit an SNMP query.
4. Creating a connection between the firewall interface and the manager.

The user must define filter rules to enable traps to go through the firewall.

A default.config file is shipped upon new installation. During the installation, no filter is activated. A pre-defined SNMP filter can be selected. If no permit on the filter rules is selected, all SNMP traffic is denied. If traffic does not match a permit rule or a denied rule, the traffic is denied.

Neither the SNMP daemon network agent (snmpd) or the SNMP firewall subagent is started during installation. Later when the subagent is started through the configuration client or SMIT and the firewall is brought down abruptly, rebooting the firewall starts the subagent automatically using the operational values given in the `/etc/security/fwsubagt.cfg` . If this file is missing, default values are used and `/etc/security/fwsubagt.cfg` is created.

Only the root authority can make changes to the Firewall Manager or starting subagent.

SNMP - Management Information Base (MIB)

See "Appendix C. SNMP Management Information Base (MIB)" on page 155 for more information on the MIB.

Chapter 5. Using the Make Key File Utility (MKKF)

If you have purchased the No Encryption version of the IBM eNetwork Firewall for AIX, this chapter does not apply.

A secure SSL network connection requires that you have:

- Configured your configuration server for SSL
- Created a key for secure communications
- Been designated as a trusted root on your server
- Stashed your key file password

Use MKKF to create the initial server key, key ring file, and certificate request. MKKF is also used to receive the initial certificate into a key ring and stash your key file password.

Creating a key file

You can create a key file for the firewall machine that can be used for both IPSEC and the configuration server.

Since the key file must be owned by the root user name, you should run this utility logged on as root.

Note: Do not give any other user or group ownership of the key file.

1. Go to the /etc/security directory and start the key utility by entering:

```
# mkkf
```

```
MKKF Key Manager  
Copyright IBM Corp. 1996  
All Rights Reserved
```

2. Create a new key ring file.

```
Key Ring Menu  
Currently Selected Key Ring: (none)
```

```
N - Create New Key Ring File  
O - Open Key Ring File  
X - Exit
```

```
Enter a command: n
```

Enter 'n' as shown above to create a new key file.

You will be prompted for a file name to use for the key file. You can use any filename, but it must end in .kyr. By default, the firewall looks for a file named fwkey.kyr.

Enter a name for the key ring file, or press ENTER to accept the default of **fwkey.kyr**

MKKF will create a new key file and display the key ring menu. Note that the key file will be listed as the currently selected key ring.

3. Create a new key and certificate request.

Key Ring Menu
Currently Selected Key Ring: fwkey.kyr

N - Create New Key Ring File
O - Open Key Ring File
S - Save Key Ring File
A - Save Key Ring as Another File
P - Set Password for Key Ring File
C - Create Stash File for Key Ring File
R - Receive a Certificate into a Key Ring File
W - Work with Keys and Certificates
X - Exit

Enter a command: **w**

Enter 'w', as shown above, to go to the Key menu.

Key Menu
Currently Selected Key Ring: fwkey.kyr
Selected Key Entry: (none)

L - List/Select a key to work with
C - Create a New Key and Certificate Request
I - Import a key from an Armored key file
X - Exit this menu

Enter a command: **c**

Enter 'c', as shown above, to create a new key.

Before a key can be stored in a key file, the key file must be password protected. MKKF will prompt you to enter a password to use to protect the key file. The password will not display when you type it. MKKF will also ask if the password should expire. Enter 'n' as shown below:

Enter password to use for the key file:
password
Enter the password again for verification: **password**
Should the password expire?
Enter Y for yes or N for no:
n
Password successfully set.
Press ENTER to continue

MKKF will prompt you for the type of key to create.

Choose Certificate Type Menu
S - Server Certificate
L - Low Assurance
C - Cancel

Enter a command: **s**

Enter 's', as shown above, to create a Server Certificate. MKKF will generate an empty certificate:

Compose Secure Server Certificate Menu

Current Certificate Information
Key Name: (none)
Key Size: 0
Server Name: (none)
Organization: (none)
Organization Unit: (none)
City/Locality: (none)
State/Province: (none)

Postal Code: (none)
Country: (none)

M - Modify the Certificate Fields
R - Ready To Create Key and Certificate Request
C - Cancel

Enter a command: m

Enter 'm' to modify the empty certificate. You will be prompted to enter information about the new certificate:

- Enter a name to use. This name can be any string and is used only by the MKKF utility:

Enter a name to use for the key entry:

Firewall Key

- Enter the size of the key. The IBM Firewall ships only the exportable version of MKKF. The maximum key size is 512.

1: 508
2: 512

Enter the number corresponding to the key size you want:

2

- Enter the fully qualified TCP/IP host name for the firewall (For example, jupiter.raleigh.ibm.com):

Enter the server's fully qualified TCP/IP domain name or press
Enter by itself to leave the field blank

jupiter.raleigh.ibm.com

- Enter an organization name to associate with the certificate. (For example, the company name):

Enter Organization Name for the certificate
or press ENTER by itself to leave the field blank.

AAA Inc.

- Enter the organizational unit name. (For example, a department name):

Enter Organizational Unit Name for the certificate
or press ENTER by itself to leave the field blank.

Network Security Products

- Enter a city where the certificate will be used:

Enter Locality/City Name for the certificate
or press ENTER by itself to leave the field blank.

RTP

- Enter a state or province.

Note: Due to the specifications for certificates, this field must be a minimum of three characters, so two-letter state abbreviations are not valid.

Enter State/Province Name for the certificate
or press ENTER by itself to leave the field blank.
State/Province must be at least three characters long.

N.C.

- Enter a postal code to associate with the certificate. (This is the same thing as a zip code):

Enter Postal Code for the certificate
or press ENTER by itself to leave the field blank.

27709

- Enter a two-letter country code:

Enter Country Code for the certificate
or press ENTER by itself to leave the field blank.
Country code must be exactly two characters long.

US

After MKKF has collected all the information from you, the certificate will be displayed:

Compose Secure Server Certificate Menu

Current Certificate Information

Key Name: Firewall Key

Key size: 512

Server Name: jupiter.raleigh.ibm.com

Organization: AAA Inc.

Organizational Unit: Network Security Products

City/Locality: RTP

State/Province N.C.

Postal Code: 27709

Country: US

M - Modify the Certificate Fields

R - Ready To Create Key and Certificate Request

C - Cancel

Enter a command: **r**

If there are any mistakes in the certificate information, you can enter 'm' to make corrections. If the information is correct, enter 'r' to create the new key and its associated key file.

MKKF will prompt you for a file to store the certificate. You can use any file name, but a good convention to follow is to use the same base name as the key file and add .cert as the extension:

Enter file to store the certificate request in:

fwkey.cert

Creating Private Key...

Private key was successfully created.

Creating certificate request...

```
certificate request was successfully created
Adding new key to key file.
The new key and certificate request were created successfully.
Press ENTER to continue
```

4. Make the newly created key the default.

After the key and certificate have been created, the Key menu will be displayed. The newly created key will be listed as the Selected Key Entry:

```
Key Menu
Currently Selected Key Ring: fwkey.kyr
Selected Key Entry: Firewall Key

L - List/Select a Key To Work With
S - Show Information about Selected Key
D - Delete Selected key
C - Create a New Key and Certificate Request
I - Import a Key From an Armored Key File
E - Export Selected Key To an Armored Key File
F - Make Selected Key the Default Key for this Key Ring
U - Unmark Selected Key's Trusted Root Status
R - Create A Certificate Request for Selected Key
X - Exit This Menu
```

Enter a command: **f**

You must make the newly created key the default key in the key file. Enter 'f' as shown in the previous example. You will be prompted to confirm the action:

```
Key Menu
Currently selected key: Firewall Key
Are you sure you want to make this key the default?
Enter Y for yes or N for No:
y
Key was made the default key.
Press ENTER to continue
```

After the key has been marked as the default, the Key Menu is displayed:

```
Key menu
Currently Selected Key Ring: fwkey.kyr
Selected Key Entry: Firewall Key

L - List/Select a Key To Work With
S - Show Information about Selected Key
D - Delete Selected key
C - Create a New Key and Certificate Request
I - Import a Key From an Armored Key File
E - Export Selected Key To an Armored Key File
F - Make Selected Key the Default Key for this Key Ring
U - Unmark Selected Key's Trusted Root Status
R - Create A Certificate Request for Selected Key
X - Exit This Menu
```

Enter a command: **x**

Exit the Key menu by entering 'x'.

5. Receive the certificate into the key ring file.

The Key Ring menu will be displayed:

```
Key Ring Menu
Currently Selected Key Ring: fwkey.kyr

N - Create New Key Ring File
O - Open Key Ring File
S - Save Key Ring File
```

A - Save Key Ring as Another File
P - Set Password for Key Ring File
C - Create Stash File for Key Ring File
R - Receive a Certificate into a Key Ring File
W - Work with Keys and Certificates
X - Exit

Enter a command: **r**

Note: Since the firewall does not use SSL for authentication purposes, your certificate does not have to be signed by a certificate authority.

Enter file name or press ENTER for Cert.txt.

fwkey.cert

This is a self-signed certificate. Add it to the key file?

Enter Y for yes or N for no:

y

Certificate added to key ring.

Press ENTER to continue

6. Create a stash file for the key file.

After the certificate has been added to the key ring, the Key Ring Menu is displayed:

Key Ring Menu
Currently Selected Key Ring: fwkey.kyr

N - Create New Key Ring File
O - Open Key Ring File
S - Save Key Ring File
A - Save Key Ring as Another File
P - Set Password for Key Ring File
C - Create Stash File for Key Ring File
R - Receive a Certificate into a Key Ring File
W - Work with Keys and Certificates
X - Exit

Enter a command: **c**

You need to create a stash file for the key file. Enter 'c' as shown in the previous example. MKKF will use the same base name as the key file name and .sth as the extension:

Stashed password file saved to fwkey.sth

Press ENTER to continue

After the stash file has been created, the Key Ring Menu is displayed:

Key Ring Menu
Currently Selected Key Ring: fwkey.kyr

N - Create New Key Ring File
O - Open Key Ring File
S - Save Key Ring File
A - Save Key Ring as Another File
P - Set Password for Key Ring File
C - Create Stash File for Key Ring File
R - Receive a Certificate into a Key Ring File
W - Work with Keys and Certificates
X - Exit

Enter a command: **x**

Your key file is now ready to be used. Enter 'x' as shown above to exit MKKF and enter 'y' to save changes to your key file as shown:

```
Key ring file has been changed. Save?  
Enter Y for yes or N for no:  
y  
Key ring saved to fwkey.kyr  
Press ENTER to continue  
#
```

7. Updating the configuration file.

After exiting the MKKF, check the file permissions on your key file, stash file and certificate file.

For security reasons, these files should be owned by root. If the files are not owned by root, change the owner using this command:

```
#ls -l fwkey*  
-rw-r--r-- 1 root security 1025 Mar 18 10:01 fwkey.cert  
-rw----- 1 root security 3682 Mar 18 10:10 fwkey.kyr4  
-rw----- 1 root security 129 Mar 18 10:09 fwkey.sth
```

After creating the key file, you must specify the key file name in the configuration server parameter file using the `fwcfsrv` command.

If you are using SSL encryption for the configuration server, you also need to set the `encryption=ssl` option using the `fwcfsrv` command.

Note: This line does not have to be changed if you are using the key file only for IPsec.

Chapter 6. Troubleshooting and Testing

This chapter tells you how to troubleshoot some of the common problems encountered when setting up and configuring the IBM Firewall. It also tells you how to test the firewall ports using the `fwice` command.

If you are having problems, first create a `firewall` log, with debug priority, to increase the information sent to your logs. See “Log File Management” on page 12 for more information.

Installation and Setup

Filter support fails

Problem Explanation

You receive these error messages.

```
Error on open/dev/ipsp_poif: A file or directory
in the path name does not exist. Filter support verification failed.
Socket creation call failed:
A file or directory in the path name does not exist.
```

This problem is caused by not rebooting the firewall after installation.

Recommended Action

Reboot your firewall and retry the procedure.

Routing Problems

The IBM Firewall provides a feature on the **Security Policy** dialog box entitled *Test IP Routing*, which can be useful for debugging routing problems. Enable this checkbox, activate your Connection configuration, and enable Connection Rules Logging. Then examine your `firewall` log to view detailed information about all packets flowing through your firewall.

Perform these tests first using IP addresses, then using host names. If your traffic routes properly using addresses but not using names, see “DNS Problems” on page 89 for more information.

Cannot ping hosts from the firewall

Problem Explanation

Your network interface is not configured properly.

Recommended Action

See your operating system documentation.

Problem Explanation

Your connection to the nonsecure network is not configured properly.

Recommended Action

Contact your Internet Service Provider for assistance.

Problem Explanation

If your secure network is isolated behind a router, your firewall must have a static route to that router. Use `netstat -rn` to verify static routing:

```
netstat -rn
```

The output should be as follows for Protocol Family 2:

Destination	Gateway	Flags
default	nrr.nrr.nrr.nrr	UG	
nnn.nnn.nnn	nnn.nnn.nnn.nnn	U	
sss.sss.sss	sss.sss.sss.sss	U	
ss1.ss1.ss1	srr.srr.srr.srr	UG	
127	127.0.0.1	U	

Figure 5. Sample output from netstat -rn.

nrr.nrr.nrr.nrr

represents your router to the internet and is the default route. The default route is a static route (Flag=UG).

nnn.nnn.nnn

represents your nonsecure domain. This is an interface route (Flag=U).

nnn.nnn.nnn.nnn

represents your nonsecure interface.

sss.sss.sss

represents your secure domain. This is an interface route (Flag=U).

sss.sss.sss.sss

represents your secure interface.

ss1.ss1.ss1

represents a subdomain on the secure side of your network and srr.srr.srr.srr represents the router to that subdomain. This is a static route (Flag=UG).

127.0.0.1

is the loopback or local host. This is an interface route (Flag=U).

You should have an interface route for each interface and your default route should point to the router on the nonsecure side of the firewall.

Recommended Action

Add a static route to your router. Contact your router administrator. Use the route add command.

Problem Explanation

The subnet mask on your secure interface or the host you are trying to contact may be incorrect.

Recommended Action

Use your client's configuration utilities to correct the mask settings.

Cannot ping nonsecure hosts from secure hosts (or vice-versa)

Problem Explanation

Each router adjacent to the firewall must contain a static route specifying the firewall as the gateway for destination networks beyond the firewall.

Recommended Action

Contact the router's administrator.

Problem Explanation

If your secure network uses addresses that are not registered and routable on the nonsecure network, including private addresses as specified in RFC 1597, packets will not be routed back to the sender.

Recommended Action

Use a client with a registered address. The firewall's NAT feature may be used for TCP and UDP traffic, but NAT will not translate addresses in ICMP packets like ping.

DNS Problems

The firewall DNS resolves names by querying the secure name server. The secure name server resolves all names in the secure network. The secure name server forwards requests for nonsecure names to the firewall name server. The firewall name server queries the nonsecure name server to resolve the request.

DNS problems can impact other areas of firewall operation. It is a good idea to check DNS even if the problem is not obviously related to DNS.

Here are some examples to lead you through each step of this method using the `nslookup` utility in order to isolate the problem. In these examples, we will use the following values:

www.ibm.com

represents an arbitrary hostname on the nonsecure network

nns.nns.nns.nns

represents the address of the nonsecure name server

sns.sns.sns.sns

represents the address of the secure name server

host.secure.company.com

represents the name of an arbitrary host inside your secure network

127.0.0.1

represents the loopback address on your firewall.

These values can be obtained from the **Domain Name Services** dialog box on the Configuration Client. You will need these values as you work through these exercises.

Note: The `nslookup` command requires the additional dot following the hostname to prevent `nslookup` from appending your secure domain name.

DNS has not been configured yet

Problem Explanation

You have not configured your firewall's DNS facilities.

Recommended Action

Complete the **Domain Name Services** dialog box.

File permissions do not allow the specified action

Problem Explanation

Firewall traffic control is not permitting the DNS packets to flow.

Recommended Action

Go to the **Security Policy** dialog box, turn on the *Permit DNS Queries* checkbox and reactivate your traffic control.

nslookup www.ibm.com. nns.nns.nns.nns fails**Problem Explanation**

The nonsecure name server is not using the indicated address or is not configured properly.

Recommended Action

Contact your DNS service provider for a valid name server address.

nslookup www.ibm.com. 127.0.0.1 fails**Problem Explanation**

named may not be running. Use `ps -ef|grep named` and look for `/usr/sbin/named`.

Recommended Action

Restart your TCP/IP services.

Problem Explanation

Your loopback adapter may be down. Use `netstat -in`; if there are asterisks behind the entry for `1n0`, loopback is down.

Recommended Action

Use `ifconfig 1n0 up` to restart your loopback adapter.

nslookup host.secure.company.com. sns.sns.sns.sns fails**Problem Explanation**

Your secure name server is down.

Recommended Action

Restart your name server.

nslookup www.ibm.com. sns.sns.sns.sns fails**Problem Explanation**

Your secure name server is not configured properly to interact with the IBM Firewall.

Recommended Action

Refer to the *IBM eNetwork Firewall User's Guide* for configuration requirements.

Configuration Client**Server not responding****Problem Explanation**

SSL encryption may not be configured properly.

Recommended Action

Ensure that SSL is selected in the client's logon panel. Use the `fwcfgsrv` command to verify that encryption is set to SSL and the SSL Key file name is specified. Also verify that the Key file and stash file are configured as described in "Chapter 5. Using the Make Key File Utility (MKKF)" on page 79 .

Problem Explanation

The firewall's configuration server may be disabled.

Recommended Action

Ensure that `/etc/inetd.conf` contains the line

```
ibmfwrcs stream tcp nowait root /usr/sbin/ibmfwrcs ibmfwrcs
```

and restart your TCP subsystems.

Problem Explanation

The firewall's configuration server may be monitoring a non-standard port.

Recommended Action

Examine `/etc/services` and ensure that it contains the line `ibmfwrcs 1014/tcp`. If you want to use the server on a different port, edit `ibmfwrcs 1014/tcp` accordingly and ensure that you specify the new port in the client's logon panel.

Problem Explanation

The firewall's traffic control may not be permitting communications to and from the Configuration Server. This only affects Configuration Clients running on a remote host.

Recommended Action

Code a connection between the machine running the Configuration Client and the firewall. The Configuration Client should be the source of the connection and the firewall the destination. Regenerate and activate your changes. See the *IBM eNetwork Firewall User's Guide* for more information.

Problem Explanation

The Configuration Server may not be configured to permit logins from a remote host.

Recommended Action

Use the `fwcfsrv` command to verify that the `localonly` parameter is set to `no`.

Unable to log on to the Configuration Server

Problem Explanation

Each user name authenticated at the firewall is configured to use any of several authentication methods. `Deny all` is used to prohibit the use of a particular service to that user.

Recommended Action

Examine the Secure Administration and NonSecure Administration fields of the username being used. These fields are only valid for Administrators, not for firewall users.

Problem Explanation

When connecting from a remote host, you get the message `Remote logins are not allowed for this account. Authentication failed.` After responding correctly to the authentication challenge, the AIX `rlogin` parameter for this account is set to `FALSE`. This happens most frequently when using the `root` user name and firewall administrators have the `rlogin` parameter set to `TRUE`.

Recommended Action

Due to security concerns, it is **not recommended** that you change the `root`'s `rlogin` setting to `TRUE`. Log onto the configuration client from the firewall console using `root`. Then create a firewall administrator and use that

ID for your remote configuration needs. Root should only be used when necessary to perform AIX administration.

Traffic Control

Filters are damaged and contact to the manager is lost.

Problem Explanation

The connection is lost between the firewall and the managed firewall.

Recommended Action

1. Logon locally to the managed firewall with the root password.
2. Select the `/etc/security/` directory.
3. Copy `fwconns.cfg.BAK` to `fwconns.cfg`.
This will put a working copy of the filter connection file in place to be activated. If problems exist other than results of adding a bad rule, you may have to copy all of `fw*.cfg.BAK` to the corresponding `cfg` file.
4. Edit `secag.cfg` and change the following two lines:
 - a. Traffic: EFM to Traffic: host
 - b. VPN: EFM to VPN: host
5. Bring up the firewall GUI and login as root in Host mode.
6. Bring up the **Connection Activation** dialog box under Traffic Control on the configuration client. Regenerate the Connection Rules from this dialog box. This will recreate and activate a working set of filters.
7. Bring up the **Virtual Private Network** dialog box under Traffic Control. Select the EFM Manager. Activate this VPN to allow the manager to regain a connection to the managed firewall.
8. From the EFM manager, force the security agreement to be transmitted and activated to the managed firewall.
9. Fix the problem that caused the connection to be lost. Transmit and activate the corrected filters.

Changes to `fwfilters.cfg` are lost

Problem Explanation

In previous releases of the IBM Firewall, filters were configured by direct edits to the filters configuration file. Starting with the IBM Firewall V3R1, the Configuration Client provides an object-oriented interface which constructs this file on your behalf. Any edits made directly to this file will be overwritten when the Configuration Client is used to make any changes.

Recommended Action

Rebuild the configuration and perform the necessary changes using the Configuration Client. See the *IBM eNetwork Firewall User's Guide* for more information.

Changes made to Connections do not take effect

Problem Explanation

Changes made to any of the Traffic Control components do not take effect until they are activated. This includes the **Security Policy** dialog box under System Administration.

Recommended Action

Use the **Connection Activation** dialog box to regenerate and activate your configuration.

Proxy Servers

No data transmitted

Problem Explanation

The firewall's proxy services are not started until the machine is rebooted after installation.

Recommended Action

Reboot the machine.

Problem Explanation

The firewall's Traffic Control must be configured to permit packets to flow to and from the proxy process, not directly through the firewall.

Recommended Action

Configure each half of the proxy connection as described in the *IBM eNetwork Firewall User's Guide*.

Use the predefined services whenever possible, particularly with FTP traffic.

Cannot connect to the desired host

Problem Explanation

If data is flowing to and from the proxy but the host cannot be contacted, your client may not be properly resolving hostnames.

Recommended Action

Ensure that *Permit DNS Queries* is enabled on the **Security Policy** dialog box and your connection configuration has been activated. See "DNS Problems" on page 89 for more information.

Problem Explanation

Each user name being authenticated at the firewall by any of the firewall services can be configured to use any of several authentication methods. Deny all is used to prohibit the use of a particular proxy to that user.

Recommended Action

Examine the user account's authentication settings in the **Users** dialog box on the Configuration Client.

Idle telnet proxy users are not being timed out

Problem Explanation

Idle proxy user timeout happens based on a command entered into the AIX crontab.

Recommended Action

Code the appropriate crontab entry. See "Appendix G. The Crontab Command" on page 191 for more information.

Network Address Translation

Ping does not work on my NAT connection

Problem Explanation

Nothing is wrong. Ping uses ICMP Echo packets. NAT does not support ICMP.

Recommended Action

None.

The NAT connection does not work

Problem Explanation

You set up and activated NAT but the connection does not work.

Recommended Action

There is either a problem with the routing tables or a NAT configuration problem.

How can a route be established for NAT packets?

Problem Explanation

There is no route established for NAT packets.

Recommended Action

Add a static route on the router in front of the firewall with the destination the NAT address(es) and the gateway the firewall.

An alternate method is to add an arp entry on the firewall for each NAT address. You might want to use the arp method while the static route is being configured.

How do I add an arp entry?

Problem Explanation

You need the following information to add an arp entry.

Recommended Action

Use the following information to add an arp entry:

MAC_ADDRESS

The hardware address of the nonsecure interface (adapter) on the firewall. To get this information use the command `netstat -in`. When using it in the arp command, change each "." to an ":".

ADAPTER_TYPE

Use ether for ethernet and 802.5 for token ring.

NAT_IP_ADDRESS

One NAT IP address.

What is the syntax for the arp command?

Problem Explanation

What is the syntax for the arp command?

Recommended Action

```
arp -s ADAPTER_TYPE NAT_IP_ADDRESS MAC_ADDRESS pub
```

How should I add the arp command?

Problem Explanation

How should I add the arp command?

Recommended Action

Use the command line to add it immediately to the arp cache. Edit the `/etc/rc.tcpip` file to have it put into the arp cache each time the firewall is booted. Either way, enter an arp command for each NAT address.

What debugging tools are available to help with NAT?

Problem Explanation

What debugging tools are available to help with NAT?

Recommended Action

NAT Logging, which allows you to trace the management of dynamic registered addresses, and IPtrace.

How do I run IPtrace?

Problem Explanation

IPtrace will show the NAT flow.

Recommended Action

If you trace on the site outside the Firewall that is the destination of the request from the secure host, you will see if the NAT translation has occurred.

Use the `-d` and `-b` options to do this. The command is `iptrace -d OUTSIDE_IP_ADDRESS -b /tmp/trace.out`. Run the test situation to and from `OUTSIDE_IP_ADDRESS`.

For example, your private address is 10.2.3.4, your NAT address is 9.45.62.108, and the Internet destination is 204.146.18.33. If you capture a direct FTP session from your secure system (10.2.3.4) to `www.ibm.com` (204.146.18.33), the iptrace would show (along with lots of other packets) the following flow:

1. scr address of 10.2.3.4 with dest address of 204.146.18.33
2. scr address of 9.45.62.108 with dest address of 204.146.18.33
3. scr address of 204.146.18.33 with dest address of 9.45.62.108
4. scr address of 204.146.18.33 with dest address of 10.2.3.4

If this flow is not complete, knowing which step is missing tells you where the problem exists.

Condition: scr/dest addresses specified in step 1 are not in trace.

Causes:

- Invalid trace command. Try a second trace with no options.
- Packets may have been blocked by filter rules. Open filters by turning on Test IP Routing (debug only) and retry IPtrace.
- Traffic not being routed to the Firewall. Use ping to verify that the secure host can get to the Firewall. Then check the routing information on the secure host and any routers between the secure host and the firewall.

Condition: scr/dest addresses specified in step 1 are found but NAT/dest specified in step 2 are not in the trace. No NAT translation has occurred.

Causes:

- There is an error in the NAT configuration. Look at the NAT configuration file and verify that the private address is in the translate list or in a map entry.
- NAT may not be active. Activate NAT.
- The Firewall interfaces are invalid for NAT. NAT only translates between secure and nonsecure interfaces (adapters). Check that the interfaces are correctly defined.
- Filters may not be active. Filters must be active for NAT. Restart the filters.

Condition: Steps 1 and 2 were found in the trace but Step 3, *NAT address as the destination*, was not found. At this point you have confirmed that NAT is working.

Causes:

- The router does not know where to send the packet. Issue an arp command to put NAT information in the arp cache on the Firewall. The command is:

```
arp -s <adapter type> <NAT IP ADDRESS>  
<MAC ADDRESS of nonsecure adapter on the Firewall> pub.
```

Your ISP needs to add routing information for NAT addresses with the Firewall as the gateway.

- The MAP function is needed. If the connection starts on the nonsecure side, a NAT map must be used.

Other General Suggestions

To see all network activity, use iptrace without any options: iptrace /tmp/trace.out. Run the test situation. When the failure has occurred, kill the iptrace.

The command: ps -ef | grep iptrace gives the process id of the iptrace.

The command: kill -9 PROCESS_ID stops the trace.

To format the information gathered use the following command: ipreport /tmp/trace.out > /tmp/trace1.

Suggested Options for Debugging NAT

- -a (without arps trace is much smaller)
- -b (need to see packets in both directions)
- -d (destination ip address)

Win 95 Remote IPsec Client

Unexpected characters received when using telnet

Problem Explanation

When connected to any host over a dialed PPP connection, not only when using the secure connection, unexpected characters are received. Some telnet clients handle the PPP protocols improperly.

Recommended Action

Get another telnet client and retry the procedure.

Log Facilities

Log facility changes do not take effect on the server

Problem Explanation

When deleting or changing a log facility, it seems to work on the GUI, but does not take effect on the server.

Recommended Action

Reboot your system.

Report Utilities

Failure on system call: ar -vt ...

Problem Explanation

The archive utility is sensitive about file names. A valid archive file name ends with .a and must be fully qualified. Empty files are allowed, but the file must exist.

Recommended Action

Provide the fully qualified name of an existing archive file.

An error occurred while accessing the file:

Problem Explanation

The above error might be seen after using any of the following commands:

```
db2 -vf fwschema.d11 > schema.out
db2 -vf fwimport.dat > import.out
db2 -vf fwqrysmpl.dml > sample.out
```

Recommended Action

Provide the correct fully qualified filenames for the .ddl, .dat, or .dml file.

Errors occur importing data to the database.

Problem Explanation

The import.out file resulting from a db2 -vf fwimport.dat>import.out command has messages that indicate one of the imports failed or was only partially successful.

Recommended Action

Check the .msg file corresponding to the import statement for which the problem was noted. It will give more detail about the problem. Look for the related record(s) in the corresponding .tbl file to see the input data and

determine what is wrong with it. For example, is it too long for its target column in the database? Is the data type appropriate for the target column type? If the input data does not look right, you might need to locate the original log file record to be sure `fwlogtbl` generated the `.tbl` file record correctly.

If you cannot resolve the problem, save the `import.out` file, the `.msg` file, the associated `.tbl` file, and the original log file before contacting IBM Service.

Log Management

`fwlogmgmt -l` fails

Problem Explanation

`fwlogmgmt -l` may generate this error message:

```
ar: 0707-106 Internal error while reading the fixed header
of archive file /foobar.a "ar" of "foobar.Z" failed in /foobar.a
Check disk space.
```

This indicates insufficient disk space in either the working directory or the destination directory.

Recommended Action

Remove the empty archive file which was created but not filed. Then increase the amount of disk space available, using the `chfs` command if necessary. See the AIX documentation for `chfs` information.

Log archival never happens

Problem Explanation

The `fwlogmgmt` command is intended to run as an AIX cron job. If the crontab has not been updated with this command, logs will not be archived automatically.

Recommended Action

Ensure you have specified appropriate log archival settings with the `fwlog` command. Then add the `fwlogmgmt` command to the AIX cron job. See "Appendix G. The Crontab Command" on page 191 for more information. You can manually add the command. It does not have to be a cron job.

Testing the Firewall Ports Using `fwice`

This section tells you how to test the firewall ports using `fwice`. When you installed the IBM Firewall, you also installed a set of test programs that you can run from workstations inside or outside the secure network to test how well the IBM Firewall protects your network. `Fwice` gives information on every port.

Use the `fwice` command to test the ports on your firewall host to see if they are responding from inside and outside the secure network. To use `fwice`, you need two files:

- The `hosts` file contains a list of host names, one for each firewall host you want to test.
- The `ports` file contains a list of the ports on those hosts that you want to test.

Running `fwice` might generate an ICA2000e message error. This might trigger a threshold violation from your log monitor facility.

The hosts file

Here is a sample entry in a hosts file:

```
124.8.7.4 test7.okla.norm.edu
```

If you do not supply a hosts file, `/etc/hosts` is used.

The services file

In the services file, each line has the format:

```
function port_no protocol
```

or

```
function port_no/protocol
```

If you do not supply a services file, the standard `/etc/services` on your system is used.

The results file

The `fwice` command stores its results in the results file you supply. If you do not supply one, the file `./results` is used. Here is a sample entry of the `./results` file.

```
9.67.96.243 tcp 7 (echo) is alive and listening.  
No connection to 9.67.96.243 on udp 7 (echo)  
9.67.96.243 tcp 9 (discard) is alive and listening.  
No connection to 9.67.96.243 on udp 9 (discard)  
No connection to 9.67.96.243 on tcp 11 (systat)  
9.67.96.243 tcp 13 (daytime) is alive and listening.  
No connection to 9.67.96.243 on udp 13 (daytime)  
No connection to 9.67.96.243 on tcp 15 (netstat)  
No connection to 9.67.96.243 on tcp 17 (qotd)  
9.67.96.243 tcp 19 (chargen) is alive and listening.  
No connection to 9.67.96.243 on udp 19 (chargen)  
No connection to 9.67.96.243 on tcp 20 (ftp-data)
```

Appendix A. Messages

This appendix contains messages for the IBM Firewall for AIX, the IBM Firewall for NT, and messages that are common to both firewalls. It also gives you the following information about the IBM Firewall messages :

- How the messages are formatted
- The messages' severity levels
- The messages and their explanations

If you have looked at a message and its explanation, but need further information, refer to "Chapter 6. Troubleshooting and Testing" on page 87.

Message Tag

- ICA** The first 3 fixed bytes.
- xxxx** A number in the range 0000 – 9999.
- a** An indicator of severity. Messages are classified by severity level.
- i – info
 - w– warning
 - e – error
 - s – severe

The numbers 0000 – 9999 are further classified into the following categories:

- 0000 – 0999 Intrusion Alarm
- 1000 – 1999 Filters
- 2000 – 2999 Proxy
- 3000 – 3999 Socks
- 4000 – 4999 Pager
- 5000 – 8999 Available
- 9000 – 9999 General/Others

Messages

ICA0001 ALERT - *count authentication failures.*

Explanation: Threshold conditions for authentication failures have been satisfied.

ICA0002 ALERT - *count authentication failures for user user_name.*

Explanation: Threshold conditions for detecting a specific log message have been satisfied.

ICA0003 ALERT - *count authentication failures from host host IP address.*

Explanation: Threshold conditions for authentication failures from any specific host have been satisfied.

ICA0004 ALERT - Tag *message_id with count log entries.*

Explanation: Threshold conditions for detecting a specific log message have been satisfied.

ICA0005 Log monitor - *out of memory.*

Explanation: Process ran out of memory.

ICA0006 Log monitor - *failure accessing services file: errno*

Explanation: Could not find entry for fwlogmond in /etc/services.

ICA0007 **Log monitor - socket creation failed:**
errno

Explanation: Could not open socket - see error message.

ICA0008 **Log monitor - bind() failed:** *errno*

Explanation: Could not bind socket - see error message.

ICA0009 **Could not open threshold definition file:** *errno*

Explanation: Problem accessing threshold definition file - see error message.

ICA0010 **Log monitor - fatal read error:** *errno*

Explanation: Problem reading from socket - see error message.

ICA0011 **Could not get status of threshold definition file:** *errno*

Explanation: Problem accessing threshold definition file - see error message.

ICA0012 **Log monitor daemon shutting down.**

Explanation: Daemon is abending or received terminate signal. Previous log messages would provide detail.

ICA0013 **Log monitor caught terminate signal.**

Explanation: Daemon received terminate signal and will shut down.

ICA0014 **Starting log monitor daemon.**

Explanation: Daemon has been started.

ICA0015 **Could not create daemon for log monitor:** *errno*

Explanation: Daemon creation failed - see error message.

ICA0016 **Could not open *process id file* - daemon may already be active.**

Explanation: Daemon could not open process id file.

ICA0017 **Could not write process id (*process id*) to file.**

Explanation: Daemon could not write process id to the file.

ICA0018 **Log monitor - empty read.**

Explanation: Received packet with no data - discarded.

ICA0019 **Log monitor - short read. Tag discarded.**

Explanation: Received packet with not enough data - discarded.

ICA0020 **Log monitor - misformatted ICA tag.**

Explanation: Received packet with misformatted data - discarded.

ICA0021 **Log monitor - misformatted authentication data.**

Explanation: Received packet with misformatted data - discarded.

ICA0022 **Invalid syntax in threshold definition file (*invalid entry*).**

Explanation: The indicated entry in the threshold file is syntactically incorrect.

ICA0023 **Can not open fwmail.conf file.**

Explanation: open on fwmail.conf file failed or file is empty

ICA0024 **Can not Connect to SMTP Server.**

Explanation: SMTP Server is busy or is refusing connection

ICA0025 **Alert Message Email failed.**

Explanation: Could not email log monitor alert message to specified address.

ICA0051 **Days to keep in log file, *log file name*, must be unsigned short integer value.**

Explanation: Days to keep in log file must be a valid integer.

ICA0052 Days to keep in archives, *log file name*, must be unsigned short integer value.

Explanation: Days to keep in archives must be a valid integer.

ICA0053 Multiple entries for the log file, *log file name*, in the *logmgmt.cfg* is not allowed.

Explanation: Multiple entries for a log file in the *logmgmt.cfg* is not allowed.

ICA0054 Can not open *\$ Variables* : file.

Explanation: Can not open *logmgmt.cfg* file.

ICA0055 There is no valid entry in *logmgmt.cfg* file.

Explanation: There is no valid entry in *logmgmt.cfg* file.

ICA0056 The log message, "*\$ Variables* :", is invalid

Explanation: The log message is invalid

ICA1001 Unable to create file with our process id

Explanation: Filter logging daemon encountered an error when writing the file *fwlogd.pid*.

User Response: Check the file system where directory */etc/security* resides. Possible out-of-space condition exists.

ICA1002 Communications with *cfgfilt* program not possible

Explanation: Due to the *fwlogd.pid* file not being created, communication between the *fwlogd* daemon and the *cfgfilt* application (required for filter control) is not possible.

User Response: Check the file system where directory */etc/security* resides. Possible out-of-space condition exists.

ICA1003 Continuing with logging daemon initialization

Explanation: The *fwlogd* daemon will continue start-up processing.

ICA1004 Filter logging daemon *fwlogd* (level *version.release*) initialized at time on date

Explanation: The IP packet logging daemon has been started. When/if packet logging is enabled daemon *fwlogd* will write the required records to the *syslog*, *local4*, file.

ICA1005 Suppressed logging of *filter_rule_no* packet message(s) due to buffer overflow

Explanation: The *fwlogd* daemon filter log buffer has overflowed. A packet for the specified filter rule cannot be logged.

User Response: Check the log. Your firewall may be under a deny-of-service attack or you may be logging messages which are not required. For example, broadcast messages should have a deny rule with log control set to no (*l=n*) to prevent filling up the log.

ICA1006 Fatal *fwlogd* error - *failing function: error message*

Explanation: The *fwlogd* server failed in the indicated function, daemon terminated.

User Response: Correct the indicated system problem and restart *fwlogd*.

ICA1007 Unable to fork child process: *errno*

Explanation: During startup of the filter logging daemon, the indicated system error was encountered.

User Response: Based on the error displayed, take corrective action.

ICA1008 Error return from *setpgrp* routine: *errno*

Explanation: During startup of the filter logging daemon, the indicated system error was encountered.

ICA1009 Unable to fork second child process: *errno*

Explanation: During startup of the filter logging daemon, the indicated system error was encountered.

ICA1010 This daemon must run with root authorization

Explanation: The filter logging daemon must be started under root authority.

User Response: Restart with root authority.

ICA1011 **sysconfig call to query kernel extension *load_path* failed: *errno***

Explanation: During startup of the filter logging daemon, the indicated system error was encountered.

ICA1012 **AIX kernel extension *netinet* not loaded -- can't continue**

Explanation: The *netinet* device driver does not contain filter support.

User Response: Install the Firewall code. Potentially, the code has been installed but the *reboot* has not been performed.

ICA1013 **Socket creation call failed: *errno***

Explanation: During startup of the filter logging daemon, the indicated system error was encountered.

ICA1014 **AIX *netinet* device driver not at required level**

Explanation: The *netinet* device driver and *fwlogd* daemon are not the same level.

User Response: Resolve the conflict, possible reboot required after installing new Firewall level.

ICA1015 **Error on *ioctl()* call (SIOCGFWLOG): *errno***

Explanation: During startup of the filter logging daemon, the indicated system error was encountered.

ICA1016 **Can't get current deferred log queue**

Explanation: Additional information associated with immediately preceding log message.

ICA1017 **Error return from SIOCGFWLOG *ioctl()* call**

Explanation: During startup of the filter logging daemon, the indicated system error was encountered.

ICA1018 **Fatal *fwlogd* error - *failing function*: *system error message***

Explanation: The *fwlogd* server failed in the indicated function, daemon terminated.

User Response: Correct the indicated system problem and restart *fwlogd*.

ICA1019 **Unexpected error exit with *rc* *internal_fw_return_code***

Explanation: During startup of the filter logging daemon, the indicated system error was encountered.

ICA1020 **Fatal *fwlogd* error - *failing function*: *return code = 0xfunction return code***

Explanation: The *fwlogd* server failed in the indicated function, daemon terminated.

User Response: Correct the indicated system problem and restart *fwlogd*.

ICA1021 **Error on open */dev/ipspp_oif*: *errno***

Explanation: The indicated device driver has not been installed.

User Response: If the Firewall code has been installed, check the */tmp/rc/net.out* file for possible error messages.

ICA1022 **Filter support verification failed**

Explanation: Due to an error recorded prior to this message, filter support cannot be verified.

ICA1023 **Error on *ioctl()* call (SIOCGFWLVL): *errno***

Explanation: During startup of the filter logging daemon, the indicated system error was encountered.

User Response: Do one of the following:

- For AIX: :p.Verify the correct level of the Firewall *netinet* device driver has been installed and the machine has been rebooted since the installation.
- For OS/390: :p.Verify the correct level of TCP/IP has been installed and has been started with the **IPCONFIG FIREWALL** configuration statement.

ICA1024 **Error writing file */etc/security/fwlogd.pid*: *errno***

Explanation: Due to the indicated system *errno*, *fwlogd* was unable to write the specified file.

User Response: Correct the indicated problem and restart the filter logging daemon.

ICA1032 Filter rules updated at *time on date*

Explanation: IP packet filtering rules have been updated.

ICA1033 Filter support (level *version.release*) initialized at *time on date*

Explanation: Firewall filter support has been initialized.

ICA1034 Filter support deactivated at *time on date*

Explanation: IP packet filtering now using default filter rules rather than those defined in */etc/security/fwfilters.cfg* file.

ICA1035 Status of packet logging set to *enabled/disabled* at *time on date*

Explanation: Status of packet logging has changed. Message indicates current state with time stamp.

ICA1036 *#:rule_noR: rule_type direction: interface s:src_addr d: dst_addr p: protocol tag: scr_port/icmp_type tag: dst_port/icmp_code r:routed/local a: secure/non_secure f:yes/no T:tunnel_id e:C/D/n l:packet_length*

Explanation: Log record indicating a processed IP packet and the corresponding filter rule it matched. For this record to be written, the matched filter rule must have log control set to *yes*. If the IP packet which matched this rule is a fragment, the ports/icmp type/code information appears for the header packet but is shown as zero for packets other than the header packet.

ICA1037 *#:rule_no action src_addr src_mask dst_addr dst_mask protocol logical_op value logical_op value interface_type routing direction!= log_control f=fragment_control!= tunnel_ID enc_alg auth_alg*

Explanation: When filters rules are updated, the activated rules are written to the log. This log message describes one of the activated rules.

ICA1038 Session Key engine started, using session socket port:*port_no* and master socket port:*port_no*

Explanation: Encryption tunnel started using specified UDP port numbers, as defined in */etc/services*.

ICA1039 Policy being (re)defined as:

Explanation: Policy cache being (re)defined using file */etc/security/fwpolicy*. Following lines show the new policy cache.

ICA1040 *>Policy statement: tunnel_origin tunnel_end tunnel_ID encrypt_flag/authenticate_flag*

Explanation: Line logged was read from the */etc/security/fwpolicy* file.

ICA1041 Context specification deleted for tunnel:*tunnel_ID*

Explanation: The tunnel context, for the listed ID, is no longer operational.

ICA1042 The following tunnel context specification(s) is defined:

Explanation: Tunnel context specifications are being defined, as listed on the following log records.

ICA1043 *>tunnel_ID:number, src_addr:IP_address, dst_addr:IP_address, encryption:algorithm*

Explanation: Message lists specific attributes of activated tunnel context.

ICA1044 Host Counter Warning: IP(IP Address) Overlimit

Explanation: There are too many secure hosts trying to connect with the Firewall machine

System Action: pass connections

ICA1045 TCP Overlimit: IP Address(Port)->IP Address(Port) rejected

Explanation: There are too many TCP sessions through the Firewall machine

System Action: reject connections

ICA1046 **UDP Overlimit:** *IP Address(Port)->IP Address(Port) rejected.*

Explanation: There are too many UDP sessions through the Firewall machine

System Action: reject connections

ICA1047 **Grace Period Warning : too many TCP sessions,***IP Address(Port)->IP Address(Port) passed*

Explanation: There are too many TCP sessions through the Firewall machine

System Action: pass connections

ICA1048 **Grace Period Warning : too many UDP sessions,***IP Address(Port)->IP Address(Port) passed*

Explanation: There are too many UDP sessions through the Firewall machine

ICA1049 **Invalid ipsec package: s:***IP Address d:IP Address protocol:Protocol spi:Security Parameters Index*

Explanation: The ipsec package cannot be decapsulated by the receiving firewall.

User Response: Ensure that the tunnel definition has been exported correctly and has been activated on each firewall.

ICA1050 **Specification deleted for tunnel:***tunnel_ID*

Explanation: The tunnel specification, for the listed ID, is no longer operational.

ICA1051 **The following tunnel specification(s) is defined:**

Explanation: Tunnel specifications are being defined, as listed on the following log records.

ICA1052 **>tunnel_ID:***number, src_addr:IP_address, dst_addr:IP_address, src_enc:algorithm rem_enc:algorithm src_mac:algorithm rem_mac:algorithm src_enc_mac:algorithm rem_enc_mac:algorithm src_pol:policy rem_pol:policy mode:transport_mode*

Explanation: Message lists specific attributes of activated tunnel.

ICA1053 **Resource allocation failure: error type is %s**

Explanation: Key Recovery encountered a problem allocating some resource. Values for number are defined as follows: 1 : Memory allocation failure. 2 : IPsec is not loaded. 3 : The tunnel does not exist. 5 : No KRB needed.

ICA1054 **Communication error: error type is %s**

Explanation: Key Recovery encountered a communications problem. Values for number are defined as follows: 1-7 : Communications problems. Check if krbpingd is running. 8-10 : Received an illegal request; it will be ignored.

System Action:

ICA1055 **SCCS internal error: error type is %s**

Explanation: An error occurred in the Key Recovery code. Values for number are defined as follows: 1-4 : Problems with crypto service provider. Have you installed SCCS? 5-7 : Problems with key recovery service provider. Have you installed KRSP? 8 : Initialization failed. 9-14 : SCCS internal error.

ICA1056 **Timeout failure: error type is %s**

Explanation: A response was not received in the allotted time. Values for number are defined as follows: 1 : Failed to receive response from remote end. Check connectivity. 2 : Failed to receive response from krbpingd. Is it running?

User Response: ----- End Key Recovery/KeyWorks-related messages. -----

ICA1200 **Terminating logging daemon due to above errors**

Explanation: Due to errors recorded prior to this message, fwlogd daemon is terminating.

System Action: IP filter logging will not be activated.

User Response: Correct indicated errors and restart fwlogd.

ICA1260 **Filter logging daemon terminating at time on date due to receipt of termination signal**

Explanation: The fwlogd daemon received the indicated termination signal and is stopping.

ICA1305 \“unknown\”

Explanation: In formatting an IP packet for syslog, a record was found with an unknown protocol specification. Protocols IP, ICMP, TCP, UDP and IPSP are the recognized protocols. Note IPSP is IBM's designation for the encrypted packets passed through a tunnel.

ICA1400 **Fatal fwtimernat error - failing function:**
system error message

Explanation: The fwtimernat server failed in the indicated function. The fwtimernat server was terminated.

User Response: Correct the indicated system problem and restart fwtimernat.

ICA1401 **Fatal fwtimernat error - failing function:**
return code = 0xfunction return code

Explanation: The fwtimernat server failed in the indicated function. The fwtimernat server was terminated.

User Response: Correct the indicated system problem and restart fwtimernat.

ICA1402 **Fatal fwtimernat error - failing function:**
error message

Explanation: The fwtimernat server failed in the indicated function. The fwtimernat server was terminated.

User Response: Correct the indicated system problem and restart fwtimernat.

ICA2000 **New FTP session to IP_address from**
IP_address (non-secure site).

Explanation: Starting a new ftp session from non-secure site.

ICA2001 **Authentication failed for user name**
(unknown) from net ftp:IP_address.

Explanation: A user, without an account, attempted to use ftp proxy from the network.

User Response: See your firewall administrator to setup a proxy account.

ICA2002 **Authentication failed for user name**
with authentication method from
network:host name.

Explanation: Firewall is unable to authenticate the indicated user name using the specified authentication method.

User Response: See your Firewall administrator.

ICA2003 **No shells configured for user name.**

Explanation: The identified user attempted a proxy login and no login shell has been defined.

User Response: See your Firewall administrator to correct this user login profile.

ICA2004 **Unknown audit event of 0xhex_value**
received.

Explanation: An unknown audit request was received by the module tcpip_audit.c.

ICA2005 **Error writing to client: errno.**

Explanation: Unable to communicate with client, see logged system message.

ICA2006 **ptelnetd: auditproc: errno.**

Explanation: Indicated error returned by telnet audit process. Potential corruption of system files.

ICA2007 **ptelnetd: panic state=value.**

Explanation: Unknown error detected. Potential corruption of system files.

ICA2008 **Non-firewall user name from**
:IP_address telneted in.

Explanation: A user, without a firewall account, attempted to use telnet proxy.

System Action: Assume Generic Authentication used.

ICA2009 **/bin/login: errno.**

Explanation: Fatal error during system login. See indicated system error message.

ICA2010 **Connect to *IP_address* from *IP_address* (non-secure).**

Explanation: Successful connection between indicated IP addresses through the non-secure interface.

ICA2011 **Connect to *IP_address* from *IP_address* (secure).**

Explanation: Successful connection between indicated IP addresses through the secure interface.

ICA2012 **New FTP session to *IP_address* from *IP_address* (secure site).**

Explanation: Starting a new ftp session.

ICA2013 **New Telnet session to *IP_address* from *IP_address*.**

Explanation: New telnet session established.

ICA2014 **Option *value* not supported.**

Explanation: The indicated flag is not supported, see preceding message.

ICA2015 **Option *-value* not supported.**

Explanation: The indicated flag is not supported, see preceding message.

ICA2016 **Remote user-id "*name*".**

Explanation: ftp connection request for indicated user.

ICA2017 **Debug - *in line*.**

ICA2018 **SNK key not found for user *name*.**

Explanation: SecureNetKey value was not found for indicated user_ID.

User Response: See your Firewall administrator for possible login configuration problem.

ICA2019 **SNK key not read properly for user *name*.**

Explanation: SecureNetKey value was not readable as octal digits for indicated user_ID.

User Response: See your Firewall administrator for possible login configuration problem.

ICA2020 ***/usr/bin/fwuserau* or */usr/bin/fwuserpt* do not exist.**

Explanation: Authentication using user-supplied authentication method is aborted.

System Action: Authentication is aborted.

User Response: Make sure that */usr/bin/fwuserau* and */usr/bin/fwuserpt* exist and the owner is the root. If the executable does not exist, user should make an executable using a compiler compatible with the operation system of the firewall and name it */usr/bin/fwuserau* or name it */usr/bin/fwuserpt*.

ICA2021 **Trying to connect to remote host *name* with user-id *name*.**

Explanation: Trying to establish a new ftp connection.

ICA2022 **Trying to connect to remote host *name*.**

Explanation: Trying to establish a new ftp connection.

ICA2023 **Usage: *ptelnetd* [-n] [-s]**

Explanation: Unknown flag specified when starting the telnet daemon.

User Response: Use only flags *-n* and/or *-s*.

ICA2024 **User *name* successfully authenticated using *method* authentication from *network:host name*.**

Explanation: FW authenticated the indicated user name using the specified authentication method.

ICA2025 **User *name* logged in using *method* authentication from *network :host name*.**

Explanation: ftp user logged in.

ICA2026 **User *name* timed out after *n* seconds at *current time*.**

Explanation: Connection attempt timed out for specified user. Potential network routing problem or remote host is not available.

ICA2027 **Connection from *remote host* at *time*.**

Explanation: Net ftp connection established to Firewall.

ICA2028 **FTP connection attempt to *IP_address* from *IP_address* refused. This machine does not support FTP from non-secure site.**

Explanation: Generally indicates an attempt to establish an ftp connection to Firewall across the non-secure interface.

System Action: Reject the connection.

ICA2029 **System error with *errno* = - in *in* line *line*.**

Explanation: The system call encounters a problem while executing a system call.

System Action: System execution halted

User Response: get the log, find out the meaning of *errno* try to resolve the problem. If cannot be resolved, contact IBM service.

ICA2030 **Function call with return code = - in *in* line *line*.**

Explanation: The function call encounters a problem.

System Action: Error returned

User Response: get the log, find out the meaning of return code try to resolve the problem. If cannot be resolved, contact IBM service.

ICA2031 **sdi function call *creadcfg()* rc = -.**

Explanation: The function call encounters a problem.

System Action: Error returned

User Response: consult the sdi reference for explanation.

ICA2032 **Lost connection.**

Explanation: Lost ftp connection.

User Response: Reestablish session.

ICA2033 **sdi function call *sd_init* rc = -.**

Explanation: The function call encounters a problem.

System Action: Error returned

User Response: consult the sdi reference for explanation.

ICA2034 **sdi function call *sd_check* rc = -.**

Explanation: The function call encounters a problem.

System Action: Error returned

User Response: consult the sdi reference for explanation.

ICA2035 **setsockopt(): *errno*.**

Explanation: System error on setsocketopt call.

ICA2036 **Telnet Session *session id* started for user *user id* (*source IP addr*:*dest IP addr*).**

Explanation: Message generated at the start of each Telnet session. A session begins when *userid*, *source ip* and *destination ip* are all known to the firewall. The *session id* is a unique identifier generated by the firewall.

ICA2037 **User *fwdfuser* or *fwdpuser* tried to login, is not allowed.**

Explanation: *fwdfuser* and *fwdpuser* are reserved users and should not be used.

System Action: Login is refused.

User Response: The administrator should investigate who is using this user.

ICA2038 **ttloop: peer died: *errno*.**

Explanation: Error occurred while flushing the network output buffer. Appears that peer process has died.

ICA2039 **ttloop: read: *errno*.**

Explanation: Error occurred while flushing the network output buffer.

ICA2040 **Authentication set to password or none is not allowed for user ID *fwdfuser*.**

Explanation: *fwdfuser* is a reserved user ID and should not use password or none as the authentication method.

System Action: Login is refused.

User Response: The administrator should change the authentication method for user ID *fwdfuser*.

ICA2041 **FTP session *session id* started for user *id* (source IP *addr*:dest IP *addr*).**

Explanation: Message generated at the start of each FTP session. A session begins when userid, source ip and destination ip are all known to the firewall. The session id is a unique identifier generated by the firewall.

ICA2042 **req_rsp_code is incorrectly set to FW_AUTH_REQ.**

Explanation: fw_tn_authenticate is not allowed to set req_rsp_code to FW_AUTH_REQ.

System Action: Abort the authentication.

User Response: Change fw_tn_authenticate, make the library fwuser.o again, and put it into the Firewall.

ICA2043 **Could not get password for *user_name*.**

Explanation: Authentication type for this user is 'password' and no password was found.

User Response: See your Firewall administrator.

ICA2044 **Incorrect time (*value*) specified for -t.**

Explanation: The time value shown contains characters outside the numeric range of 0..9 or exceeds the maximum allowed value.

ICA2045 **Option -T not supported on firewall.**

Explanation: Indicated option is not supported.

ICA2046 **Option -k not supported on firewall.**

Explanation: Indicated option is not supported.

ICA2047 **Option -s not supported on firewall.**

Explanation: Indicated option is not supported.

ICA2048 **Option -u not supported on firewall.**

Explanation: Indicated option is not supported.

ICA2049 **Unknown flag -*value* ignored.**

Explanation: Indicated flag was specified and is not recognized.

ICA2050 **Unknown parm *value*.**

Explanation: Indicated value, specified as an option, is not recognized.

ICA2051 **adapt_addr conversion error on address.**

Explanation: IP address shown is not valid.

User Response: Possible corruption of the file /etc/security/fwsecadpt.cfg. Remove the file, reconfigure your secure interface(s) and reinitialize the filters.

ICA2052 **afopen failed to open /etc/security/login.cfg: *errno*.**

Explanation: Unable to authenticate user, open error on indicated file.

ICA2053 **Could not open secure interface file.**

Explanation: A secure interface has not been configured.

User Response: If a secure interface should be defined, use Firewall commands/smit panels to define the secure interface(s).

ICA2054 **enduserdb rc=*value*, *errno*.**

Explanation: Received indicated system error code attempting to retrieve user login profile information.

User Response: See your Firewall administrator to verify your login account.

ICA2055 **getpeername() (*invocation name*): *errno*.**

Explanation: System error when ftp daemon attempted to get socket name.

ICA2056 **getsockname() (*invocation name*): *errno*.**

Explanation: System error when ftp daemon attempted to get port name.

ICA2057 **getuser non-secure shell rc=*value* for *user_ID*, *errno*.**

Explanation: Received indicated system error code attempting to retrieve shell name for connection from non-secure side of Firewall.

User Response: See your Firewall administrator to set a shell for your user login profile.

ICA2058 **getuser secure shell rc=value for user_ID, errno.**

Explanation: Received indicated system error code attempting to retrieve shell name for connection from secure side of Firewall.

User Response: See your Firewall administrator to see a shell for your user login profile.

ICA2059 **ioctl(): errno**

Explanation: System error on ioctl() call for SIOCSPGRP.

ICA2060 **ptelnetd: ftok for shared memory failed.**

Explanation: Unable to allocate shared memory segment.

User Response: Contact the Firewall administrator, apparent memory problem.

ICA2061 **ptelnetd: shmat for shared memory failed.**

Explanation: Unable to allocate shared memory segment.

User Response: Contact the Firewall administrator, apparent memory problem.

ICA2062 **ptelnetd: shmget for shared memory failed.**

Explanation: Unable to allocate shared memory segment.

User Response: Contact the Firewall administrator, apparent memory problem.

ICA2063 **setsockopt() (SO_DEBUG): errno.**

Explanation: Indicated error message returned from system call 'setsockopt'.

ICA2064 **setsockopt() (SO_KEEPALIVE): errno.**

Explanation: Indicated error message returned from system call 'setsockopt'.

ICA2065 **setuser rc=value, errno.**

Explanation: Received a bad return code on a system call for the indicated reason.

ICA2066 **signal(): errno.**

Explanation: System error when ftp daemon attempted to establish signal handler.

ICA2067 **Fatal pftpd initialization error - bind(): errno**

Explanation: pftpd server initialization failed, daemon terminated.

User Response: Correct the indicated system problem and restart pftpd. The most likely cause of this error is another ftp daemon already listening on the standard ftp port (21).

ICA2068 **Fatal pftpd initialization error - listen(): errno**

Explanation: pftpd server initialization failed, daemon terminated.

User Response: Correct the indicated system problem and restart pftpd.

ICA2069 **Fatal pftpd error - main accept(): errno**

Explanation: pftpd server main routine failed, daemon terminated.

User Response: Correct the indicated system problem and restart pftpd.

ICA2070 **Fatal pftpd initialization error - socket(): errno**

Explanation: pftpd server initialization failed, daemon terminated.

User Response: Correct the indicated system problem and restart pftpd.

ICA2071 **Connection refused, maximum number of connections reached.**

Explanation: The pftpd server cannot create another FTP session because the maximum number of sessions already exist.

System Action: The connection is refused.

User Response: Wait for existing connections to end, then try the request again.

ICA2072 **ftp configuration file (*filename*) is not available.**

Explanation: ftp daemon attempted to open the specified ftp configuration file but it either does not exist or could not be opened.

System Action: ftp daemon processing uses the default configuration

User Response: None, unless the file should exist, in which case it should be created or moved to the location specified in the message.

ICA2073 **Unable to obtain storage for ftp language table.**

Explanation: Storage required to represent a REPLYLANGUAGE statement in the ftp configuration file could not be obtained.

System Action: Processing continues.

User Response: Increase the region size or reduce the entries in the configuration file.

ICA2074 **Processing complete for ftp config statement: *configuration statement***

Explanation: ftp has processed the indicated configuration statement.

System Action: Processing continues.

User Response: None

ICA2075 **FTP for *user id (source IP addr:dest IP addr)*, operation file name, numbytes bytes. sid: *session id*.**

Explanation: Message generated for each file transfer on open FTP sessions. The sid is a unique identifier generated by the firewall at session start.

ICA2076 **FTP Session *session id* ended for user *id (source IP address:dest IP addr)*, duration seconds, numbytes bytes.**

Explanation: Message generated at the end of each FTP daemon session. The sid is a unique identifier generated by the firewall at session start.

ICA2077 **Telnet Session *session id* ended for user *id (source IP address:dest IP addr)*, numbytes bytes.**

Explanation: Message generated at the end of each Telnet session. The sid is a unique identifier generated by the firewall at session start.

ICA2078 **Disconnected proxy user *user* - idle for *time* minutes.**

Explanation: User's session has exceeded maximum allowable idle time.

ICA2079 **Attention - Unauthorized connection attempt to *IP_address* from *IP_address*.**

Explanation: Generally indicates an attempt to establish a connection to Firewall across the non-secure interface.

System Action: Reject the connection.

ICA2080 **Syntax error (*reason*) near column *column* in ftp configuration file line *line*: *configuration statement***

Explanation: The ftp configuration statement at the given line is in error. The reason for the error and the location where the error was detected is provided.

System Action: Statement is ignored.

User Response: Correct the statement in the ftp configuration file.

ICA2081 **No message catalog given by ftp configuration statements is usable.**

Explanation: Attempts to open the message catalogs given by the REPLYLANGUAGE ftp configuration statements failed. No client message catalog can be used.

System Action: Client message catalog is forced to the English language in the C directory.

User Response: Ensure that there are catalog files in each of the directories associated with the language directories in the ftp configuration REPLYLANGUAGE statements. Also check that the NLSPATH environment variable is correctly set to allow substitution of both the sub-directory from the LANG environment variable (%L) and the catalog name (%N).

ICA2082 **Unable to set ftp LANG environment variable to *sub-directory*, reason: *reason***

Explanation: A system error (given by the reason) occurred when the ftp daemon was trying to change the setting of the LANG environment variable to the sub-directory specified.

System Action: Processing continues. Recovery may generate other messages.

User Response: Use the reason given to determine if this is a system error or programming error.

ICA2083 **Unable to open ftp client message catalog in directory:** *sub-directory*,
reason: *reason*

Explanation: ftp daemon could not open the message catalog in the given sub-directory. The reason given is the errno returned from catopen().

System Action: Processing continues. Recovery may generate other messages.

User Response: Ensure that there is a catalog in the directory associated with the language directory provided. Check that the NLSPATH environment variable is correctly set to allow substitution of both the sub-directory (%L) and the catalog name (%N).

ICA2084 **Forcing ftp client message catalog to English via the C sub-directory.**

Explanation: Due to previously listed errors, the ftp daemon has forced the client message catalog to the English language using the C sub-directory.

System Action: If the language can be forced to the C message catalog processing continues. If it can not, the program exits.

User Response: Correct the error from the previous messages. If the program also existed, create the message catalog in the C sub-directory and set the NLSPATH environment variable correctly.

ICA2085 **Telnet Session ended for pid** *Process id*
(source IP address).

Explanation: Message generated at the end of each Telnet session.

ICA2086 **Misconfigured user file; user** *user* **with no key** (*key*).

Explanation: ftpd found requested user in user file, but could not find key - misconfigured user file.

User Response: use Firewall commands/smit panels to correct this problem.

ICA2087 **ftpd could not find the specified user**
user in the user config file.

Explanation: the username specified has not been configured or the user.cfg file is corrupt.

User Response: use Firewall commands/smit panels to correct this problem.

ICA2088 **ftpd could not open user configuration file.**

Explanation: ftpd made a call to fopen which failed because it could not open the user config file.

User Response: Make sure the user config file (user.cfg by default) is available; use Firewall commands/smit panels

ICA2089 **Authorization type from user file**
(Authorization type) did not match any
entries in table (struct tab2 authtabY).

Explanation: The authorization type of the specified user (returned from user.cfg) does not match any supported types (such as deny,none,sdi,password,etc.)

User Response: Check user.cfg file integrity or configuration; use Firewall commands/smit panels to correct this problem.

ICA2090 **Authentication failed for user** '*user name*' **from** *client ip* **because**
KEY=DENY in the user.cfg file.

Explanation: Authentication failed due to user.cfg file specifications set by the Firewall administrator.

User Response: See your Firewall administrator.

ICA2091 **User** '*user name*' **not allowed to ftp to**
the non-secure port (*firewall ip*).

Explanation: User tried to ftp into the firewall server via a non-secure port (nsp) - all nsp users must have their 'fwnsftp' key properly configured to a valid authorization type (in the user.cfg file).

User Response: Check user.cfg file integrity or configuration; use Firewall commands/smit panels to correct this problem.

ICA2092 **Internal Error: nt_gwauth() failed.**

Explanation: nt_gwauth() normally returns one of three values (AUTHENTICATED,NOT_AUTHENTICATED or DENY) in this \ case nt_gwauth returned some invalid integer.

ICA2093 **User** '*user name*' **not allowed to ftp to**
the secure port (*port number*).

Explanation: User tried to ftp into the firewall server via a secure port (sp) - all sp users must have their 'fwsftp' key properly configured to a valid authorization type (in the user.cfg file).

User Response: Check user.cfg file integrity or configuration; use Firewall commands/smit panels to correct this problem.

ICA2094 **Login Failed: expected format: "PASS <password>" after: "USER <user name>"; received invalid cmd.**

Explanation: Authentication failed because the ftp client did not send the expected format (PASS 'password' per RFC959)

User Response: Type "user <username>"; enter correct password. See your Firewall administrator.

ICA2095 **Login Failed: (via method *auth method*) failed authentication of user '*user name*' from client ip (client site).**

Explanation: Authentication failed due to an invalid input (by client for specified authentication type) - such as user entered invalid password, etc.

User Response: See your Firewall administrator.

ICA2096 **Authenticated: (via method *auth method*) successful authentication of user '*user name*' from client ip (client site).**

Explanation: Authentication succeeded

ICA2097 **httpd --> Starting HTTP proxy server version *HTTP Proxy Version*.**

Explanation: HTTP Proxy for WWW access starting.

ICA2098 **httpd --> Shutting down HTTP proxy server.**

Explanation: HTTP Proxy for WWW access shutting down.

ICA2099 **httpd --> Status: <*HTTP Status code*> from client <*IP address*>, who requested <"*HTTP GET request*"> for <*number of bytes*> bytes.**

Explanation: Status of client HTTP request for some file thru the proxy. For further information about the "Status" code value, see the HTTP 1.0(RFC 1945) or HTTP 1.1(RFC 2068) documents (or superceding RFCs) available at various sites on the internet, including ds.internic.net.

ICA2100 **Socket address equals zero.**

Explanation: An invalid destination address was found in the local request.

ICA2101 **Socket address family error: *sin_family_type*.**

Explanation: An invalid address family type was found in the local request.

ICA2102 **Error initializing odm: *odmerrno*.**

Explanation: An odm_initialize() error occurred for ODM (Object Data Manager).

ICA2103 **Error setting odm default path: *odmerrno*.**

Explanation: An odm_set_path() error occurred for ODM (Object Data Manager). object class, OCSvhost.

ICA2104 **Error locking odm database: *odmerrno*.**

Explanation: An odm_lock() error occurred for ODM (Object Data Manager).

ICA2105 **Error opening odm object *Customized_Attribute: odmerrno*.**

Explanation: An odm_open_class() error occurred for ODM (Object Data Manager).

ICA2106 **Error searching odm object *OCS_virtual_host: odmerrno*.**

Explanation: An odm_get_first() error occurred for ODM (Object Data Manager). object class, OCSvhost.

ICA2107 **Error closing odm object *OCS_virtual_host: odmerrno*.**

Explanation: An odm_close_class() error occurred for ODM (Object Data Manager). object class, OCSvhost.

ICA2108 **Error unlocking odm database: *odmerrno*.**

Explanation: An odm_unlock() error occurred for ODM (Object Data Manager).

ICA2109 **Error terminating odm:** *odmerrno*.

Explanation: An `odm_terminate()` error occurred for ODM (Object Data Manager).

ICA2110 **Error getting server by name:** *errno*.

Explanation: An `getservbyname()` error occurred. The host Login Monitor service, `lm`, is not specified properly in the `/etc/services` file.

ICA2111 **byname() error:** *errno*.

Explanation: An `gethostbyname()` error occurred. The host machine name is not specified properly in `/etc/hosts`.

ICA2112 **Invalid protocol name:** *protocol_name*.

Explanation: The protocol name specified in the ODM object class, `OCSvhost`, is not supported.

ICA2113 **Error opening socket to LM:** *errno*.

Explanation: A `socket()` error occurred to host machine where the Login Monitor resides.

ICA2114 **Error binding local address:** *errno*.

Explanation: A `bind()` error using the local address for this OCS node.

ICA2115 **Error connecting socket to LM:** *errno*.

Explanation: A `connect()` error occurred to the host machine where the Login Monitor resides.

ICA2116 **Protocol type error:** *protocol_type*.

Explanation: The virtual terminal protocol type used to communicate with the host Login Monitor is invalid.

ICA2117 **Malloc error on LM message.**

Explanation: A `malloc()` error occurred when dynamically allocating space for the variable-length Login Monitor message.

ICA2118 **Error transmitting msg to LM:** *errno*.

Explanation: A `send()` error occurred when sending Login Monitor a request to open the correct host device.

ICA2119 **Error receiving msg from LM:** *errno*.

Explanation: A `recv()` error occurred when Login Monitor returns an acknowledgement.

ICA2120 **Status error from LM:** *status*.

Explanation: The acknowledgement from Login Monitor indicates that host device was NOT successfully opened.

ICA2121 **Error opening OCS administration device:** *errno*.

Explanation: The OCS administration device was not successfully opened.

ICA2122 **Failed covering IP address to TBM ID:** *errno*.

Explanation: `ioctl()` `OCS_GET_TBMID` error occurred. `ioctl` command `OCS_GET_TBMID` failed on the OCS administration device.

ICA2123 **Error Connecting TBM determined by rlogin:** *errno*.

Explanation: `ioctl()` `OCS_IS_TBM_CONNECTED` error occurred. `ioctl` command `OCS_IS_TBM_CONNECTED` failed on the OCS administration device.

ICA2124 **No host nodes are connected:** *errno*.

Explanation: There are no host nodes connected to this OCS node from the list of possible host nodes.

ICA2125 **Error getting list for ODM(Object Data Manager):** *Customized_Attribute: odmerrno*.

Explanation: An `odm_get_list()` error occurred for ODM object class, `CuAt(Customized Attribute)`.

ICA2126 **No OCS host node name associated with:** *hostnode_to_connect*.

Explanation: The `CuAt(Customized Attribute)` entry was found but there was no `hostnode/ocsnode` match.

ICA2127 **Malloc error on Host array.**

Explanation: A `malloc()` error occurred when dynamically allocating space for the array of possible host names.

ICA2128 **User (unknown) from *client ip* (client site) attempted a command '*invalid command*' before authentication.**

Explanation: A user attempted actions before entering in username and password for authentication - users must first be authenticated before any further processing may continue.

User Response: Please login with USER and PASS

ICA2129 **gethostbyname (*invocation name*): *errno***

Explanation: System error when ftpd attempted to get host information corresponding to the host name.

ICA2130 **User (*username*) from *client ip* (client site) attempted a command '*invalid command*' .**

Explanation: Specified user attempted invalid command.

User Response: Only commands USER, QUOTE SITE and QUIT are allowed until you specify "quote site destination".

ICA2131 **Authentication failed for user '*user name*' from *client ip* because of an error in the user.cfg file.**

Explanation: Authentication failed due to a user.cfg file specifications set by the Firewall administrator (check previous logs).

User Response: See your Firewall administrator.

ICA2132 **User '*user*' from ip *client ip* (client site) attempted the invalid command '*invalid command*' .**

Explanation: The user attempted an invalid command. The only valid commands at this point are SITE,USER, and QUIT.

ICA2133 **Error: *function call failed in instance:line, WSAGetLastError***

Explanation: General error message; check logs

ICA2134 **Notice: ftpd: connect() (in *instance*) could not reach IP, WSAGetLastError.**

Explanation: Connect() could not find the requested address; check WSAGetLastError result.

User Response: double-check your address - may be DNS or network error

ICA2135 **Data transfer completed: Received *bytes bytes* (from *source IP*); sent *bytes bytes* (to *destination IP*).**

Explanation: This information reflects a single data transfer during a particular ftp session. \ However, note that it is possible that the data transfer may not have successfully completed \ (check log for a failed recv or send call).

ICA2136 **Error: CreateThread() failed in *instance: errno*.**

Explanation: ftpd could not create a thread

ICA2137 **Data connection established; server: *source ip* client: *destination ip*.**

Explanation: Successful data connection.

ICA2138 **Insufficient memory: pftpd: malloc(*bytes*) returned NULL in function *instance*.**

Explanation: Unable to allocate enough memory - malloc returned NULL.

ICA2139 **LogonUser() failed: *reason*.**

Explanation: The Windows NT (SAM) API LogonUser (for password authentication) failed due to specified reason(s).

User Response: Contact the Firewall administrator.

ICA2140 **httpd --> HTTP Proxy authentication result for user < *user*>, on < *user ip*>, thru network ... RC:< *reason*>.**

Explanation: The HTTP Proxy attempted user authentication. It's success or failure is reported here for the specified reason.

User Response: Contact the Firewall administrator.

ICA2141 **FTP session to *IP_address* from *IP_address* terminates.**

Explanation: The ftp session to firewall terminates.

ICA2142 **fw_tn_authenticate authenticated *userid* successfully.**

ICA2143 fw_tn_authenticate authentication for *userid* failed.

Explanation: fw_tn_authenticate cannot authenticate the specified user ID.

System Action: Login is refused.

User Response: If fw_tn_authenticate has any logging facilities, then the administrator should look at the log file to determine the cause.

ICA2144 fw_tn_authenticate did not return successfully.

Explanation: The value returned by fw_tn_authenticate is not zero. The function in fw_tn_authenticate might be missing.

System Action: Login is refused.

User Response: Look at fw_tn_authenticate carefully to see if it ever returns a non-zero value and correct it if it occurs. If that is the case, make the library fwuser.o again and put it into the Firewall.

ICA2145 The system returned return code *rc* in file *filename* at line *linenumber*.

Explanation: A system call failed. The library fwuser.o might be absent.

System Action: Authentication is aborted.

User Response: Make sure that /usr/lib/fwuser.o is present. If it is, contact your IBM representative.

ICA2146 The IBM-supplied fwuser.o has not been replaced.

Explanation: You are using the IBM-supplied fwuser.o because you have not replaced it with your own fwuser.o.

System Action: Authentication is aborted.

User Response: You should write and compile your own authentication if you defined any user to use User-Supplied authentication. The IBM-supplied fwuser.o denies access to all non-AIX and non-Firewall users.

ICA2147 fwtnet: user *user id* started a transparent telnet session from *source IP addr* (secure side) to *dest IP addr*.

Explanation: Message generated at the start of each transparent proxy session (fwtnet). A session begins when *userid*, *source ip* and *destination ip* are all known to the firewall. Only session started from secure side is allowed.

System Action: allow the transparent telnet.

ICA2148 Attention -- Unauthorized connection attempt for user *user id* from *source IP addr* (nonsecure side) to *dest IP addr*, is not allowed.

Explanation: Generally indicates an attempt to establish a connection to Firewall across the non-secure interface.

System Action: Reject the connection.

User Response: You should telnet from secure side using transparent proxy.

ICA2149 fwtnet: a LOGIN_ADAPTER_ERROR occurred while starting a transparent telnet session from *source IP addr* to *dest IP addr*.

Explanation: A LOGIN_ADAPTER_ERROR occurred when calling q_check_secure(0).

System Action: Reject the connection.

User Response: check the secure adapter.

ICA2150 Pftpd error - failing function: return code = *0xfunction return code*

Explanation: The pftpd server detected an error in the indicated function. The daemon terminates.

User Response: Correct the indicated system problem and restart pftpd.

ICA2151 Login refused.

Explanation: This message is to be displayed to user who tries to login but not allowed.

ICA2152 fwlogin: write to *device* failed.

Explanation: Cannot write to the device.

ICA2153 fwlogin: read from *device* failed.

Explanation: Cannot read to the device.

ICA2154 error in *portname* with *reason*.

Explanation: This Firewall encountered a problem.

ICA2155 Pftpd error - failing function: system error message

Explanation: The pftpd server detected an error in the indicated function. The daemon terminates.

User Response: Correct the indicated system problem and restart pftpd.

ICA2156 Attention -- User *user id* tried to use transparent ftp from NONSECURE side source IP addr to dest IP addr , was not allowed.

Explanation: Generally indicates an attempt to establish a connection to Firewall across the non-secure interface.

System Action: Reject the connection.

User Response: You should ftp from secure side using transparent proxy.

ICA2157 User *user id* from source IP addr is not allowed to use transparent proxy to dest IP addr.

Explanation: Generally indicates an attempt to establish a connection to Firewall while transparent proxy is not configured.

System Action: Reject the connection.

User Response: turn fwtp proxy ftp = on

ICA2158 Option *value* was specified incorrectly.

Explanation: Indicated flag was specified incorrectly.

ICA2159 Timeout value not specified for -t option.

Explanation: A timeout value must be supplied for the -t option.

ICA2160 Password changed for user *user ID* from *network :host name*.

Explanation: An FTP user has successfully changed his password in the password database.

System Action: None

User Response: None

ICA2161 User *user ID* attempted login using expired password from *network :host name*.

Explanation: An FTP user attempted to establish a connection to the Firewall using an expired password.

System Action: The FTP login validation fails and the user is returned to the FTP command shell.

User Response: The user must attempt to validate again through the FTP USER command or by re-establishing the FTP connection and passing the password string of the form "old_password/new_password/new_password".

ICA2162 Password change failure for user *user ID* from *network :host name*.

Explanation: An FTP user attempted to change his password and the password validation routine failed. The possible reasons for the failure include: - Incorrect "old" password was specified, - Only one occurrence of the "new" password was specified, - Two occurrences of "new" password do not match, or - Delimiter used to separate passwords was not "/".

System Action: FTP password validation fails and the user is returned to the FTP command shell.

User Response: Attempt to re-validate with the FTP server verifying the passwords are being entered correctly. If the problem persists, contact the service representative.

ICA2163 safemaid started.

Explanation: Starting safemaid.

ICA2164 safemaid stop.

Explanation: stopping safemaid.

ICA2165 Interrupted telnet session.

Explanation: Telnet session is ending, but it cannot retrieve its session information from the pipe. The session was probably interrupted during startup by the client, thus the session was not fully initialized.

ICA2166 Could not retrieve attribute *attribute* for user *user id*. Return code = *return code*.

Explanation: The authentication service could not retrieve the specified attribute from the user database for the specified user. System Action : The user authentication fails.

User Response: Contact system administrator to correct the user's database record.

ICA2167 *user id authentication failed for service using authentication scheme from client address on network type*

Explanation: The specified user failed to be authenticated for the specified service using the specified authentication method. The user was requesting the service from the indicated address and network type. System Action : The user authentication fails.

User Response: Contact system administrator.

ICA2168 *user id authentication failed for service due to storage shortage.*

Explanation: User ID could not be authenticated for service because there was a memory allocation failure during authentication processing. System Action : The user authentication fails.

User Response: Contact system administrator.

ICA2169 **User *name* successfully authenticated for service using *method* from *network:host name*.**

Explanation: FW authenticated the indicated user name for the requested service using the specified authentication scheme.

ICA2170 *user id authentication failed for service. auth method is not registered with the Firewall.*

Explanation: User ID could not be authenticated for service. The requested authentication method is not registered with the Firewall. System Action : The user authentication fails.

User Response: Contact system administrator.

ICA2171 **Account *user_name* has been locked due to an expired password.**

Explanation: The password has expired and not been changed. This account has been locked.

System Action: The account is locked and Firewall password authentications will fail. UserRes

ICA2172 **Account *user_name* is locked.**

Explanation: This account has been locked.

System Action: The account is locked. Firewall password authentications will fail.

User Response: See your Firewall administrator for unlocking the account.

ICA2173 **User tried to login using reserved user name *user id*.**

Explanation: The ID supplied by the user is reserved for use by the firewall.

System Action: Login is refused.

User Response: The administrator should investigate who is using this username.

ICA2174 *user id authentication failed for service using authentication scheme from client address on network type due to an internal processing error.*

Explanation: The specified user failed to be authenticated for the specified service using the specified authentication method. The user was requesting the service from the indicated address and network type. The authentication request failed due to an internal processing error. System Action : The user authentication fails.

User Response: Contact system administrator.

ICA2175 **Windows NT LogonUser call failed for user *user name*. Last error was *last error*.**

Explanation: The specified user name failed to be authenticated by the Windows NT LogonUser API call. Windows NT reported last error after LogonUser failed. System Action : The user authentication fails.

User Response: Contact system administrator.

ICA2176 **Unknown authentication scheme *authentication scheme* was defined for user name using component from *network*.**

Explanation: The specified authentication scheme was defined for the specified user when using the specified firewall component from the specified network but the authentication scheme is not currently registered with the firewall. System Action : The user authentication request fails.

User Response: Contact system administrator.

ICA2177 **SafeMail connection *0xsession ID* received from *socket peer name*.**

Explanation: SafeMail received an inbound connection from the peer name listed. The indicated connection ID number has been assigned for tracking purposes. (Debug level)

System Action: A thread has been dispatched to handle this connection.

ICA2178 **SafeMail session 0xsession ID has been established from sender's IP address to recipient's IP address.**

Explanation: SafeMail has established contact with the recipient mail server and is ready to transfer mail. (Info level)

System Action: Data transfer is about to begin.

ICA2179 **SafeMail has forwarded message size bytes for connection 0xsession ID from sending server's address to receiving server's address.**

Explanation: SafeMail has successfully forwarded a message between the two mail servers listed. This session was previously identified in an ICA2166 message. This message contained the number of bytes indicated. (Info level)

ICA2180 **SafeMail terminated session 0xSession ID from sender's address.**

Explanation: SafeMail has refused to transfer the mail being sent in the indicated session. (Info level)

System Action: The session has been terminated.

User Response: Increase the logging priority level to obtain more detailed diagnostic information.

ICA2181 **SafeMail terminated session 0xSession ID for reason code reason code.**

Explanation: SafeMail's main processor terminated the indicated session because a primary error condition was detected. Reason codes include: 01 - unable to locate the recipient mail server 02 - sender attempted to route mail between two nonsecure servers 03 - recipient mail server rejected the connection, may be down 04 - recipient mail server refused to accept the mail 05 - one or more connections timed out; either the sending or the receiving mail server may be down 06 - recv() returned 0 bytes; either the sending or the receiving mail server may be down 07 - recv() returned negative; either the sending or the receiving mail server may be down 08 - too many error commands were received 09 - select() return negative; either the sending or the receiving mail server may be down This message is logged at Debug level.

System Action: The connection has been terminated.

ICA2182 **SafeMail rejected session 0xSession ID because of an invalid SMTP command command, reason code reason code.**

Explanation: SafeMail's command-validation subroutine detected an invalid or a dangerous command. These reason codes vary for each SMTP command. See the IBM Firewall Support web page for current values. (Debug level)

System Action: The connection has been terminated.

User Response: Correct the sending mail client or the sending mail server so that safe and valid information is being sent.

ICA2183 **httpd --> HTTP Proxy Configuration file (filename) is not available.**

Explanation: The HTTP proxy daemon attempted to open the specified configuration file but it either does not exist or could not be opened.

System Action: HTTP Proxy does not start

User Response: Configure the proxy via the GUI or the fwhttp command and restart the proxy.

ICA2184 **signal() error with signal signal No.. safemaid exit.**

Explanation: System error when safemaid daemon attempted to establish signal handler.

ICA2185 **Cannot open socket. safemaid exit**

Explanation: Failure while opening the socket.

ICA2186 **Cannot bind the socket to the port. safemaid exit**

Explanation: SafeMail could not bind to the appropriate port. The port number specified in the services file may be out of range or not valid, or there may be another service already using same port. SystemAction:SafeMail has ended. UserResponse:Correct the services file, or reconfigure the other application to use a different port.

ICA2187 **Cannot accept new connection. safemaid try again**

Explanation: Failure while accepting new connection.

ICA2188 **Incorrect time (*value*) specified for -l.**

Explanation: The time value shown contains characters outside the numeric range of 0..9 or exceeds the maximum allowed value.

ICA2189 **Timeout value not specified for -l option.**

Explanation: A timeout value must be supplied for the -l option.

ICA2190 **SafeMail could not use the indicated cache file directory (*Cache file*). SafeMail has terminated.**

Explanation: Safemail needs some space on the indicated file system for storing temporary data. This directory must exist and it must be writeable by the "nobody" user. The default directory is /tmp.

System Action: Safemail has stopped.

User Response: The -f parameter should be used to select a different directory, or the new directory should be created, or the file system permissions should be corrected.

ICA2191 **The error occurred writing the SafeMail cache file for session *Session ID*.**

Explanation: Safemail needs some space on the indicated file system for storing temporary data.

System Action: In the event of a delivery problem, SafeMail would be unable to send a failure notification. Safemail will continue to attempt delivery, and successful deliveries will not be impeded.

User Response: Increase the amount of free space in the cache directory, or select a different location with more space available.

ICA2192 **Session *Session ID* from secure sender *IP Address* identified itself with a nonsecure domain name, *Sender's domain*.**

Explanation: The indicated session is receiving a note from a secure network address, but the sender's ID which was sent in the MAIL FROM command does not match any of the configured secure mail domains. This is often caused by misconfigured mail clients.

System Action: Message delivery will be attempted anyway, but in the event of a delivery failure, SafeMail will not be able to report that failure to the sender.

User Response: Reconfigure the sending client to send its private domain name.

ICA2193 **Session *Session ID* encountered an error attempting delivery to *IP Address*.**

Explanation: SafeMail was unable to deliver the message to the indicated mail server.

System Action: Delivery will continue to all other recipients. At the end of the transmission, the sender will be notified of all failures.

User Response: No further action is necessary.

ICA2194 **A mail routing loop has been detected from *Sending server* to *Receiving server* for domain name.**

Explanation: The specified mail exchanger for the indicated domain was either the same as the sender or same as one of the Firewall's interfaces.

System Action: SafeMail refused this recipient. Delivery will continue to any additional recipients.

User Response: Examine the mail routing architecture and implementation, including the mail-exchanger information stored in DNS, for possible loops.

ICA2200 **(*service:function*) WinSocket initialization error : *WSAGetLastError***

Explanation: Error occurred when initializing WinSocket.

User Response: Correct the system problem indicated by *WSAGetLastError* and restart the indicated service (First Parameter).

ICA2201 **(*service:calling function*) failed function failed at line *line number* : *WSAGetLastError***

Explanation: The Networking component specified has failed

User Response: Correct the system problem indicated by *WSAGetLastError* and restart the indicated service (First Parameter).

ICA2202 **(*service:calling function*) timeout timed out after *WSAGetLastError* seconds :**

Explanation: The indicated function timed out after idling for the specified time.

User Response: Reconnected to the indicated service and respond before the indicated timeout

ICA2203 (service:calling function) **Memory error; failed function returned return value at line line number: WSAGetLastError**

Explanation: Memory error has occurred, usually out of memory; check WSAGetLastError

User Response: Free up disk space - consult System Administrator

ICA2204 (service:calling function) **filename error: access denied or creation failed.**

Explanation: The indicated service encountered an error when attempting to access or create the specified file or the file associated with the file parameter.

User Response: Make sure the indicated filename exists and has the correct permissions.

ICA2205 (service:calling function) **File filename is required but could not be found.**

Explanation: The file specified does not exist. The most likely reason for the failure is that the Firewall default configuration was erased. Restore the file from a current backup.

User Response: Verify that the configuration file does not exist. The configuration program expects this file to exist. If a backup version is not available contact your service representative.

ICA2206 (service:calling function) **Configuration file filename is corrupted.**

Explanation: The indicated configuration file is not in a usable format. The contents have become corrupted. The most likely reason for the corruption is that the file was manually edited and invalid data added.

User Response: The configuration file will need to be recreated correctly. First cat the file (or make a viewable copy) then erase the original file. Reconfigure the file by using the appropriate firewall configuration command using the original file for reference, if necessary.

ICA2207 (service:calling function) **Configuration file filename is empty.**

Explanation: The indicated configuration file was either not found or the file was found, but it is empty. The most likely reason for the file not being found is that the configuration for the indicated service has not been performed.

User Response: Verify the state of the configuration file. If the file exists, the configuration command expects this file to contain data. Consult the manual for additional information.

ICA2208 service **Session session id started for user id from a non-secure adapter (source IP address:dest IP addr).**

Explanation: Message generated at the beginning of each indicated session.

ICA2209 service **Session session id ended for user id from a non-secure adapter (source IP address:dest IP addr); bytes total bytes.**

Explanation: Message generated at the end of each indicated session. Total Bytes indicates the number of bytes transferred during the session. Services (i.e., ptnetd) that do not support Total Bytes will indicate zero.

ICA2210 (service) **User user id attempted login using expired password from source IP address (non-secure).**

Explanation: The indicated user attempted to establish a connection to the Firewall using the indicated expired password from the indicated source IP on a non-secure adapter.

User Response: The password given has expired per password ruleset. Contact your system admin.

ICA2211 (service) **User user id attempted login using expired password from source IP address (secure).**

Explanation: The indicated user attempted to establish a connection to the Firewall using the indicated expired password from the indicated source IP on a secure adapter.

User Response: The password given has expired per password ruleset. Contact your system admin.

ICA2212 (service) **User name was successfully authenticated from source IP address (secure).**

Explanation: FW authenticated the indicated user name from the indicated source IP on a secure adapter.

ICA2213 (service) **User name was successfully authenticated from source IP address (non-secure).**

Explanation: FW authenticated the indicated user name from the indicated source IP on a non-secure adapter.

ICA2214 (service) **User name failed authentication from source IP address (non-secure).**

Explanation: FW failed authentication for the indicated user name from the indicated source IP on a non-secure adapter.

User Response: Most likely cause was incorrectly typed user name or password; User names and passwords are case sensitive (check Caps Lock).

ICA2215 (service) **User name failed authentication from source IP address (secure).**

Explanation: FW failed authentication for the indicated user name from the indicated source IP on a secure adapter.

User Response: Most likely cause was incorrectly typed user name or password; User names and passwords are case sensitive (check Caps Lock).

ICA2216 (service) **User name from source IP address (non-secure) did not enter matching (verification) passwords.**

Explanation: A password change was requested or required and the indicated user from the indicated source IP on a non-secure adapter entered passwords that did not match. The user authentication data was not changed.

User Response: Changing passwords requires typing the password twice, the second time for verification; Most likely cause was an incorrectly typed verification password.

ICA2217 (service) **User name from source IP address (secure) did not enter matching (verification) passwords.**

Explanation: A password change was requested or required and the indicated user from the indicated source IP on a secure adapter entered passwords that did not match. The user authentication data was not changed.

User Response: Changing passwords requires typing the password twice, the second time for verification; Most likely cause was an incorrectly typed verification password.

ICA2218 service **Session session id started for user id from a secure adapter (source IP address:dest IP addr).**

Explanation: Message generated at the beginning of each indicated session.

ICA2219 service **Session session id ended for user id from a secure adapter (source IP address:dest IP addr); bytes Total Bytes.**

Explanation: Message generated at the end of each indicated session. Total Bytes indicates the number of bytes transferred during the session. Services (i.e., ptnetd) that do not support Total Bytes will indicate zero.

ICA2220 (service) **User user id started a transparent proxy session from source IP addr (secure side) to dest IP addr.**

Explanation: Message generated at the start of each transparent proxy session .A session begins when userid, source ip and destination ip are all known to the firewall. Only session started from secure side is allowed.

System Action: allow the transparent proxy.

ICA2221 (service) **Warning: IP (Control IP addr) at peer end of Control line was not equal to IP (Data IP addr) at peer end of Data line.**

Explanation: For Security purposed (i.e., anti-hijacking) Make sure the IP Address of the the peer to which the Control Connection socket is connected is the same as the IP of the peer to which the Data Connection socket is connected. These may be different if using Net Dispatcher or if the destination has used multiple adapters

System Action: Check to see if the Destination FTP Server is using multiple adapters or Net Dispatcher is being used. Make sure filters only allows valid IP addresses through port 20 and port 21.

ICA2222 (service) **Warning! Protocol violation. Received Non-RFC compliant command invalid string; Expected protocol string.**

Explanation: The indicated service received an unexpected string which is not compliant with the associated RFC; possible hacker.

System Action: Use a Client that complies with the RFC for the indicated service

ICA3001 ***Alert*: real user is *ident* user name, not socks connect user name**

Explanation: Possible security breach attempt, user name not authenticated.

ICA3006 **count bytes from *client*, count bytes from *server***

Explanation: Message indicating number of bytes transferred between the sockd daemon and its respective client and server hosts.

ICA3007 **A connection was refused due to exceeding the maximum connection count.**

Explanation: The socks server is configured to only accept a certain maximum number of client sessions. This message is generated when that threshold has already been met and additional connection requests arrive.

System Action: The newly-attempted connection is closed.

User Response: The maximum number of concurrent connections is determined by the SOCKS5_MAXCHILD parameter in socks5.conf. Increase this setting and refresh the server. See the IBM Firewall reference for details. start unused

ICA3010 **connected -- Bind from *user(real_user)@src_addr* for *dst_addr* (*destination port*)**

Explanation: Connection established.

ICA3011 **connected -- Connect from *user(real_user)@src_addr* to *dst_addr* (*application*)**

Explanation: Successful socket connection to outside world.

ICA3012 **refused -- Connect from *user(real_user)@src_addr* to *dst_addr* (*application*)**

Explanation: Remote host refused connection.

ICA3013 **select() *errno***

Explanation: System error.

ICA3014 **terminated -- Bind from *user(real_user)@src_addr* for *dst_addr* (*destination port*).(*count bytes from client*, *count bytes from server*)**

Explanation: Connection terminated.

ICA3015 **terminated -- Connect from *user(real_user)@src_addr* to *dst_addr* (*destination host*).(*count bytes from client*, *count bytes from server*)**

Explanation: Connection to server terminated.

ICA3016 *****Cannot find appropriate interface to communicate with *destination host***

Explanation: File /etc/sockd.route does not contain routing information for the specified destination host.

ICA3017 **Cannot execute shell command for pid *sockd process***

Explanation: Sockd daemon unable to execute a /bin/sh command.

User Response: Verify the /bin/sh shell is available on the system.

ICA3018 **refused -- Bind from *user(real_user)@src_addr* for *dst_addr***

Explanation: Remote host refused connection.

ICA3019 **Error in GetDst() from host *socks_src_name: errno***

Explanation: Error in resolving destination address for requested connection.

ICA3022 **Invalid != field at line *line number***

Explanation: Invalid entry found in /etc/sockd.conf file.

ICA3023 **Invalid comparison at line *line number***

Explanation: Invalid entry found in /etc/sockd.conf file.

ICA3024 Invalid entry at line *line number*

Explanation: Invalid entry found in /etc/sockd.route file.

ICA3025 Invalid permit/deny field at line *line number*

Explanation: Invalid entry found in /etc/sockd.conf file.

ICA3026 Invalid port number at line *line number*

Explanation: Invalid entry found in /etc/sockd.conf file.

ICA3027 Shell Command Failed (*exec status*) for "*cmd*"

Explanation: Displayed shell command failed.

User Response: Verify shell processor is available on the system.

ICA3030 Unable to open config file (*/etc/sockd.conf*)

Explanation: Open request against indicated file failed.

ICA3031 Unable to open routing file (*/etc/sockd.route*): *errno*

Explanation: Open request against indicated file failed.

User Response: See your Firewall administrator. A default file was provided during Firewall installation.

ICA3032 Unable to open userfile (*user name file*): *errno*

Explanation: The filename specified for *=userlist on a permit rule could not be found.

ICA3033 Unexpected result from Validate()

Explanation: Identd verification of the user name was specified, Identd responded with unexpected result.

ICA3035 Cannot connect to identd on *client host*

Explanation: Identd verification of the user name was specified, Identd does not respond.

ICA3039 Error -- shell command "*cmd*" contains no alphanumeric characters.

Explanation: Invalid shell command, see log message.

ICA3040 Error -- shell_cmd fork() *errno*

Explanation: Sockd daemon unable to switch to child process via 'fork()'

ICA3041 Error -- unable to get client address.

Explanation: Error return from 'getpeername()' call.

User Response: Check routing and DNS configuration.

ICA3042 Error -- undefined command (*0xhex-command-received*) from host *client address*

Explanation: Invalid command received from client application.

User Response: Possible client configuration problem, or mismatch on client and Firewall support level.

ICA3043 Error -- wrong version (*0xhex-version-number*) from host *client address*.

Explanation: Firewall supports socks version 4.2.

User Response: Possible client configuration problem, or mismatch on client and Firewall support level.

ICA3044 Failed -- Connect from *user(real_user)*@*src_addr* to *dst_addr* (*application*). Error code: *command causing failure errno*.

Explanation: Connection request failed.

ICA3045 Failed -- Bind from *user(real_user)*@*src_addr* for *dst_addr*. Error: connected to wrong host *dst_name* (*dst_port* (*application*)).

Explanation: Bind request failed.

ICA3046 **Failed -- Bind from**
user(real_user)@src_addr for dst_addr.
Error code: *command causing failure*
errno.

Explanation: Bind request failed.

ICA3047 **Timed-out -- Bind from**
user(real_user)@src_addr for dst_addr

Explanation: Connection timed out.

ICA3048 **Shell command too long:** *command...*

Explanation: The command to be executed, from the /etc/sockd.conf file, is too long.

ICA3049 **Timed-out -- Connect from**
user(real_user)@src_addr to dst_addr
(application)

Explanation: Connection timed out.

ICA3050 *matched sockd.conf filter rule*

Explanation: Filter rule from the /etc/sockd.conf file which matched the socks connection.

ICA3051 **AIX sockd_route() cannot find interface**
for remote address.

Explanation: Could not find interface route information.

ICA3052 **Error setting userid to "nobody".**

Explanation: Could not set userid of the child sockd process to "nobody".

ICA3053 **Error on popen(AIX route script):**
system error message

Explanation: Failure running script to find routing information.

ICA3054 **Fatal memory allocation failure in AIX**
sockd_route().

Explanation: Memory allocation failure trying to gather routing information.

ICA3055 **Fatal error AIX sockd_route() parsing**
for first space in: *input line*

Explanation: Error parsing system route information.

ICA3056 **Fatal error AIX sockd_route() parsing**
for second space in: *input line*

Explanation: Error parsing system route information.

ICA3057 **Fatal error in AIX sockd_route()**
reading route script output: *system*
error message

Explanation: Error reading script output.

ICA3058 **Error on popen(AIX adapter script):**
system error message

Explanation: Failure running script to find interface information.

ICA3101 **Sockd error sending data - select():**
system error message

Explanation: (SOCKS422) Error while sending data.

ICA3102 **Sockd error sending data - write():**
system error message

Explanation: (SOCKS422) Error while sending data.

ICA3103 **Sockd error receiving data - select():**
system error message

Explanation: (SOCKS422) Error while receiving data.

ICA3104 **Sockd error receiving data - read():**
system error message

Explanation: (SOCKS422) Error while receiving data.

ICA3105 **Cannot create process id file filename.**

Explanation: (SOCKS422) Process id file creation/write failed.

ICA3106 **Sockd failed to fork child:** *system error*
message

Explanation: (SOCKS422) Attempt to fork child to handle a SOCKS request failed.

ICA3107 **Set inbound socket SO_LINGER option failed:** *system error message*

Explanation: (SOCKS422) not critical

ICA3108 **Set outbound socket SO_LINGER option failed:** *system error message*

Explanation: (SOCKS422) not critical

ICA3109 **Invalid entry at line *line number* in file *filename*.**

Explanation: (SOCKS422) Incorrect configuration entry syntax.

ICA3110 **Illegal interface field at line *line number* in file *filename*.**

Explanation: (SOCKS422) Incorrect configuration entry syntax.

ICA3111 **Illegal destination IP at line *line number* in file *filename*.**

Explanation: (SOCKS422) Incorrect configuration entry syntax.

ICA3112 **Illegal destination mask at line *line number* in file *filename*.**

Explanation: (SOCKS422) Incorrect configuration entry syntax.

ICA3113 **Parsed *number of lines* lines in file *filename*.**

Explanation: (SOCKS422) Incorrect configuration entry syntax.

ICA3114 **No valid lines found in file *filename*.**

Explanation: (SOCKS422) Configuration file empty, or incorrect syntax.

User Response: Correct the indicated configuration file.

ICA3115 **Invalid 'permit/deny' field at line *line number* in file *filename*.**

Explanation: (SOCKS422) Incorrect configuration entry syntax.

ICA3116 **Invalid '?=' field at line *line number* in file *filename*.**

Explanation: (SOCKS422) Incorrect configuration entry syntax.

ICA3117 **Illegal source IP at line *line number* in file *filename*.**

Explanation: (SOCKS422) Incorrect configuration entry syntax.

ICA3118 **Illegal source mask at line *line number* in file *filename*.**

Explanation: (SOCKS422) Incorrect configuration entry syntax.

ICA3119 **Invalid comparison at line *line number* in file *filename*.**

Explanation: (SOCKS422) Incorrect configuration entry syntax.

ICA3120 **Invalid port number at line *line number* in file *filename*.**

Explanation: (SOCKS422) Incorrect configuration entry syntax.

ICA3121 **Received SIGUSR1 - dumping socks configuration.**

Explanation: (SOCKS422) Signal to dump active configuration to log file, following this message.

ICA3122 **Socketd could not fork daemon: *system error message***

Explanation: (SOCKS422) Fork to initialize socketd daemon failed.

User Response: Correct the indicated system problem and restart socketd.

ICA3123 **Socketd server starting.**

Explanation: (SOCKS422) Socketd has successfully initialized and is awaiting connections.

ICA3124 **Fatal sockd initialization error - bind():**
system error message

Explanation: (SOCKS422) Sockd server initialization failed, daemon terminated.

User Response: Correct the indicated system problem and restart sockd.

ICA3125 **Fatal sockd initialization error - listen():**
system error message

Explanation: (SOCKS422) Sockd server initialization failed, daemon terminated.

User Response: Correct the indicated system problem and restart sockd.

ICA3126 **Fatal sockd error - main accept():**
system error message

Explanation: (SOCKS422) Sockd server main routine failed, daemon terminated.

User Response: Correct the indicated system problem and restart sockd.

ICA3127 **Sockd server received terminate**
signal.

Explanation: root or nobody killed the process, daemon terminated.

User Response: Restart sockd if the administrator so desires (type "sockd").

ICA3128 **Fatal sockd initialization error -**
socket(): *system error message*

Explanation: Sockd server initialization failed, daemon terminated.

User Response: Correct the indicated system problem and restart sockd.

ICA3129 **Fatal sockd initialization error - failing**
function: *system error message*

Explanation: Sockd server initialization failed in the indicated function, daemon terminated.

User Response: Correct the indicated system problem and restart sockd.

ICA3130 **Sockd error - failing function:** *system*
error message

Explanation: The sockd server detected an error in the indicated function. The daemon continues, but connections may be refused or terminated.

User Response: If the problem persists, stop sockd, correct the indicated system problem and restart sockd.

ICA3131 **Error reading file name. Previously**
cached data will be used.

Explanation: The file could not be read or contained incorrect data. A previous message should describe the problem. Sockd will continue to operate with cached data from the previous version of the file.

User Response: Correct the error in the indicated file.

ICA3132 **Unknown flag -value.**

Explanation: The indicated flag is not recognized, daemon terminated.

User Response: Correct the syntax and restart sockd.

ICA3133 **Unknown parameter value.**

Explanation: The indicated parameter is not recognized, daemon terminated.

User Response: Correct the syntax and restart sockd.

ICA3134 **Conflicting options option1 and option2.**

Explanation: The indicated options cannot be specified together, daemon terminated.

User Response: Correct the syntax and restart sockd.

ICA3135 **Sockd error - failing function: return**
code = 0x*function return code*

Explanation: The sockd server detected an error in the indicated function. The daemon terminates.

User Response: Correct the indicated system problem and restart sockd.

ICA3700 **WinSocket initialization error :**
WinSocket error

Explanation: Error occurred when initializing WinSocket.

User Response: Correct the indicated system problem and restart sockd.

ICA4000 *program - Warning: Received signal signal, terminating ...*

Explanation: Termination due to receipt of signal.

ICA4001 **STOP** *program as PID processId*

Explanation: Prints end of daemon completion. Informational message.

ICA4002 **Temporary ID**

Explanation: Informational message.

ICA4003 **Problem with child process** *processId.*

Explanation: Could not create a child process.

ICA4004 **Fatal Error. Killing fwpagerd on signal signal.**

Explanation: Signal handler.

ICA4005 **No fwpagerd daemon running, program not found.**

Explanation: Could not send a page as daemon was not active.

ICA4006 **No fwpagerd daemon running with process id processId.**

Explanation: Could not find the process Id of the daemon process.

ICA4007 **START** *program as PID processId*

Explanation: Print start information. Informational message.

ICA4008 **Cannot set sigignore for SIGPIPE.**

Explanation: Failure while setting up to ignore the broken pipe signal.

ICA4009 **Cannot set sigset for SIGCHILD.**

Explanation: Failure while setting up to catch a dying child signal.

ICA4010 **Cannot set termination process.**

Explanation: Failure while setting signal to catch termination process.

ICA4011 **Cannot open socket.**

Explanation: Failure while opening socket.

ICA4012 **Cannot set sigset for SIGTERM.**

Explanation: Failure while setting up to catch SIGTERM & SIGINT signals.

ICA4013 **Cannot set socket reuse option.**

Explanation: Failure while setting socket reuse option.

ICA4014 **Cannot set socket linger option.**

Explanation: Failure while setting socket linger option.

ICA4015 **Cannot bind the socket to the port.**

Explanation: Failure while binding the socket to the port.

ICA4016 **Cannot set listen on socket.**

Explanation: Failure while setting up to listen on socket.

ICA4017 **Service servName using TCP socket socket.**

Explanation: Informational msg.

ICA4018 **Function call select() failed.**

Explanation: Internal function call failure.

ICA4019 **Severe error from new_work().**

Explanation: Internal severe error from new_work routine.

ICA4020 **Error(program): Could not write to stream socket: socket**

Explanation: Possible system error.

User Response: Check socket usage.

ICA4021 Problem receiving response.

Explanation: Problem receiving response from modem.

User Response: Check modem connections and the initialization string.

ICA4022 Request successful.

Explanation: Informational message.

ICA4023 Request failed.

Explanation: Request to send page has failed.

ICA4024 Error(program): Priority out of range (minpri - maxpri).

Explanation: Incorrect priority range.

User Response: Correct priority range. Valid values are from -1 through 5.

ICA4025 Error(program): Address must be in the form of ID@carrier when -n option is used.

Explanation: Incorrect command usage syntax.

User Response: Correct command usage syntax.

ICA4026 Error(program): Unknown host hostname

Explanation: Could not resolve hostname.

User Response: Check hostname.

ICA4027 Error(program): Could not open stream socket : errno

Explanation: Could not create a new socket.

ICA4028 Error(program): Could not set socket options : errno

Explanation: Could not set socket linger option.

ICA4029 Error(program): Could not connect to host : errno.

Explanation: Could not connect to the host.

User Response: Check serial port configuration and existence of device driver file.

ICA4030 Error(program): Could not write to stream socket : errno.

Explanation: Could not write to the stream socket.

ICA4031 Problem receiving response. Condition of message unknown.

Explanation: Problem receiving response from modem.

ICA4032 Message sent successfully to queue.

Explanation: Informational message. Message has been sent to queue.

ICA4033 Message failed. No message(s) sent.

Explanation: Could not send the message onto the pager queue.

ICA4034 date Failed (ID ID Pri priority Secs period Tries retryCount) [fromEntry] personName: message.

Explanation: Displays this message when the page is sent unsuccessfully.

ICA4035 Cannot re-queue message mesg from program to person.

Explanation: Could not send into paging queue.

ICA4036 SUCCEEDED (ID ID Pri priority Secs period Tries retryCount) [fromEntry] personName: message.

Explanation: Displays this message when the page is sent successfully. Informational message.

ICA4037 DUMPED to dumpFile (ID ID Pri priority Secs period Tries retryCount) [fromEntry] personName: message.

Explanation: Pages that are not sent immediately are dumped to a file to be tried later.

ICA4038 Cannot write to dump file dumpFile.

Explanation: Dump file cannot be written into.

User Response: Check file system permissions.

ICA4039 **lpcKey: 0x***lpcKey*

Explanation: Informational message.

ICA4040 **Retry time of** *retryTime* **minutes exceeded.**

Explanation: Failed to initialize modem after the specified minutes.

User Response: Check initialization string.

ICA4041 **Found alphanumeric message for numeric pager.**

Explanation: Numeric pagers cannot contain alphanumeric data.

User Response: Correct using smitty/SMIT menu.

ICA4042 **Person cannot receive pages.**

Explanation: Pager is probably not activated.

User Response: Check pager for activation.

ICA4043 **Carrier** *carrier* **does not exist.**

Explanation: Carrier specified does not exist.

User Response: Correct using smitty/SMIT menu.

ICA4044 **Carrier** *carrier* **does not have a DTMF phone number.**

Explanation: Carrier specified does not have the DTMF phone number.

User Response: Correct using smitty/SMIT menu.

ICA4045 **Pager number** *pageNumber* **is too long for carrier's maximum of** *carrLen*.

Explanation: Pager number is too long for carrier's maximum.

User Response: Use another shorter pager number less than that of the carrier's maximum.

ICA4046 **Pager number** *pageNumber* **is too long for default length of** *defaultCarrLen*.

Explanation: This message occurs when the default length is too less.

User Response: Correct using smitty/SMIT menu. Increase default length.

ICA4047 **Problem at line** *lineNumber* **of modem file** *ModemfilePathname*.

Explanation: Modem definition file contains an invalid character.

User Response: Correct using smitty/SMIT menu.

ICA4048 **Cannot open modem on device** */dev/deviceName*.

Explanation: Could not open modem on specified device.

User Response: Check or re-configure serial port. Check device.

ICA4049 **Modem open on** */dev/deviceName*.

Explanation: Informational message. Modem has been successfully detected on the serial port.

ICA4050 **Cannot set modem characteristics.**

Explanation: Failed while trying to set modem characteristics.

User Response: Check modem initialization string.

ICA4051 **Cannot initialize modem after** *numInitTries* **retries.**

Explanation: Modem could not be initialized.

User Response: Check modem initialization string and serial port configuration.

ICA4052 **Cannot dial pager number** *pageNumber*

Explanation: Pager number cannot be dialed.

User Response: Check pager number validity.

ICA4053 **Cannot hangup modem.**

Explanation: Cannot hangup modem.

User Response: Check modem initialization string and hangup command used.

ICA4054 **Cannot dial message** *message*

Explanation: Cannot dial message.

ICA4055 **Problem at line *lineNumber* in modem file *filename*.**

Explanation: Invalid modem definition file.

User Response: Correct using smitty/SMIT menu.

ICA4056 **Cannot dial carrier *carrier*'s DTMF number (*DTMFnumb*).**

Explanation: DTMF number may have been changed or is incorrect for this carrier.

User Response: Correct using smitty/SMIT menu.

ICA4057 **Cannot transmit block.**

Explanation: Failed while trying to transmit block.

User Response: Check carrier parameters using smitty/SMIT menu.

ICA4058 **No response to transmitted block.**

Explanation: Could not get a response from the carrier after transmitting block.

User Response: Check carrier parameters using smitty/SMIT menu.

ICA4059 **Cannot receive response to message delivery.**

Explanation: Could not get a response from the carrier after message delivery.

User Response: Check carrier parameters using smitty/SMIT menu.

ICA4060 **Cannot transmit pager id.**

Explanation: Cannot transmit pager id.

User Response: Check pager number and carrier parameters using smitty/SMIT menu.

ICA4061 **Cannot transmit end <CR> of automatic mode request.**

Explanation: Cannot transmit end <CR> of automatic mode request.

User Response: Check carrier parameters using smitty/SMIT menu.

ICA4062 **Cannot transmit automatic mode request.**

Explanation: Cannot transmit automatic mode request signal.

User Response: Check carrier parameters using smitty/SMIT menu.

ICA4063 **Failed to receive go-ahead from carrier *carrier* after *numTries* retries.**

Explanation: Carrier may be busy at this time.

User Response: Check carrier parameters using smitty/SMIT menu and try later.

ICA4064 **Communications error during prompt with carrier *carrier*.**

Explanation: Communications error may occur for a number of reasons. Try again later.

User Response: Check carrier parameters using smitty/SMIT menu and try later.

ICA4065 **Cannot receive response to logon.**

Explanation: Modem cannot receive response to logon.

User Response: Check modem initialization string and carrier parameters.

ICA4066 **Carrier *carrier* did not respond to logon attempt.**

Explanation: Carrier did not respond to logon attempt.

User Response: Check carrier parameters using smitty/SMIT menu and try later.

ICA4067 **Carrier *carrier* said *receiveDataString*.**

Explanation: Carrier transmitted back some error message or busy message.

User Response: Check carrier parameters using smitty/SMIT menu and try later.

ICA4068 **Carrier *carrier* forced a disconnect during logon.**

Explanation: Carrier forced a disconnect during logon.

User Response: Check carrier parameters using smitty/SMIT menu.

ICA4069 **Dumping messages to carrier** *carrier*
caused by *ConnectRetryMax* **retry**
loops.

Explanation: If carrier is busy, the program dumps pages and tries later.

ICA4070 **Skipping messages to carrier** *carrier*
caused by *maxTotalTries* **session**
connect tries.

Explanation: Carrier cannot be contacted after a number of tries.

User Response: Check carrier parameters and try again later.

ICA4071 **Error(program): Cannot allocate**
memory for carrier retry: *errno*.

Explanation: Possible system or memory allocation errors.

ICA4072 **Error(program): Cannot add to carrier**
retry list: *errno*.

Explanation: Carrier possibly may not exist.

User Response: Check carrier validity and try again.

ICA4073 **Data connection to carrier** *carrier* **at**
phoneNumber failed after *retryCount*
retries.

Explanation: Data connection has failed.

User Response: Check modem connections and carrier parameters using smitty/SMIT menu.

ICA4074 **ID prompt from carrier** *carrier* **was not**
received after *numTries* **retries.**

Explanation: Carrier failed to response with an ID or acknowledgement prompt.

User Response: Make sure carrier uses the TeleAlphanumeric Protocol.

ICA4075 **Communications error during logon**
with carrier *carrier*.

Explanation: Communications error could occur for a number of reasons.

User Response: Check carrier parameters using smitty/SMIT menu.

ICA4076 **Maximum logon attempts to carrier**
carrier exceeded.

Explanation: Carrier has failed to respond within the specified attempts.

User Response: Check carrier parameters and try again later.

ICA4077 **Message go-ahead not received from**
carrier *carrier*.

Explanation: Carrier has failed to response with a go-ahead prompt.

User Response: Check carrier parameters and try again later.

ICA4078 **Cannot create blocks.**

Explanation: Carrier could not create blocks for transmission.

User Response: Check carrier parameters using smitty/SMIT menu.

ICA4079 **Carrier** *carrier* **did not respond to**
message delivery.

Explanation: Carrier had trouble delivering the message.

User Response: Check carrier parameters using smitty/SMIT menu.

ICA4080 **Carrier** *carrier* **forced a disconnect**
during message delivery.

Explanation: Carrier forced a disconnect during message delivery.

User Response: Check carrier parameters and modem initialization string.

ICA4081 **Carrier** *carrier* **rejected message or**
Pager ID.

Explanation: Carrier rejected the pager message or pager id.

User Response: Check validity of pager id, activation of pager and carrier parameters.

ICA4082 **Communications error during message delivery to carrier** *carrier*.

Explanation: Communications errors could occur for a number of reasons.

User Response: Check carrier parameters using smitty/SMIT menu.

ICA4083 **Failed to receive confirmation from carrier** *carrier* after *maxTries* retries.

Explanation: This message occurs if the carrier is busy or cannot establish a connection.

User Response: Check carrier parameters using smitty/SMIT menu and try again after a few minutes.

ICA4084 **Cannot transmit <EOT>.**

Explanation: Modem cannot transmit <EOT>.

User Response: Check modem connections and initialization string.

ICA4085 **Cannot receive response to <EOT>.**

Explanation: Modem cannot receive response to <EOT>.

User Response: Check modem connections and initialization string.

ICA4086 **Carrier** *carrier* **did not respond to <EOT>.**

Explanation: Carrier cannot respond to transmitted data.

User Response: Check carrier validity and modem connections.

ICA4087 **Carrier** *carrier* **responded with data unacceptable error because of contents.**

Explanation: Carrier cannot respond to transmitted data.

User Response: Check carrier parameters using smitty/SMIT menu.

ICA4088 **Cannot open defaults file** *defaultPathname*.

Explanation: The modem defaults file may not exist or has incorrect permissions.

User Response: Check file for existence and permissions.

ICA4089 **Incomplete defaults file** *defaultPathname*.

Explanation: The modem defaults file has missing data.

User Response: Correct using smitty/SMIT menu.

ICA4090 **Invalid outside line number in defaults file** *defaultPathname* at line *lineNumber*.

Explanation: Carrier database file has an invalid outside line number.

User Response: Clean the carrier database file.

ICA4091 **Invalid baud rate value in defaults file** *defaultFile* at line *lineNumber*.

Explanation: Carrier database file has an invalid baud rate.

User Response: Clean the carrier database file.

ICA4092 **Invalid data bit value in defaults file** *defaultFile* at line *lineNumber*.

Explanation: Carrier database file has an invalid data bit value.

User Response: Clean the carrier database file.

ICA4093 **Invalid parity value in defaults file** *defaultFile* at line *lineNumber*.

Explanation: Carrier database file has an invalid parity value.

User Response: Clean the carrier database file.

ICA4094 **Invalid stop bit value in defaults file** *defaultFile* at line *lineNumber*.

Explanation: Carrier database file has an invalid stop bit value.

User Response: Clean the carrier database file.

ICA4095 **Unrecognized tag** *tag id* **in defaults file** *defaultFile* **on line** *lineNumber*.

Explanation: Carrier database file has an invalid tag.

User Response: Clean the carrier database file.

ICA4096 **Incorrect number of parameters.**

Explanation: Informational message.

ICA4097 **Error(program): Cannot create carrier list. Memory problems.**

Explanation: Possible system or memory problems.

ICA4098 **Error(program): Errors in paging carrier file *carrierFile*.**

Explanation: Carrier database file has some invalid data.

User Response: Check the carrier database file for invalid tags.

ICA4099 **Error(program): Cannot get IPC token *errno*.**

ICA4100 **Error(program): Cannot create retry list. Possible memory problems.**

Explanation: Possible system error or memory problems.

ICA4101 **Error(carrier): Cannot create queue, page_q_err: *pageQErr*.**

ICA4102 **Error(program): Cannot setup signal catch for SIGTERM/SIGINT: *errno*.**

Explanation: Possible system error.

ICA4103 **Error(program): Cannot set modem characteristics for carrier *carrier*.**

Explanation: Could not setup the modem.

User Response: Check serial port configuration and initialization string.

ICA4104 **Missing tag *tag* for carrier *carrier*.**

Explanation: Missing modem information. A tag could be baud rate, outside line, etc..

User Response: Check modem configuration file for invalid characters.

ICA4105 **Carrier *carrier* must have at least one phone number listed.**

Explanation: Carrier must contain the phone number.

User Response: Add the phone number using smitty/SMIT menu.

ICA4106 **Cannot open file *CarrierFileName*.**

Explanation: Carrier database file must exist.

User Response: If not already present, create one using smitty/SMIT menu.

ICA4107 **Line *lineNumber* too long.**

Explanation: Line in carrier database file is too long.

User Response: Check carrier database file for invalid line.

ICA4108 **Unknown tag at line *lineNumber*.**

Explanation: Unknown tag exists in carrier database file.

User Response: Check carrier database file for invalid tag.

ICA4109 **Invalid sequence at line *lineNumber*.**

Explanation: Invalid sequence exists in carrier database file.

User Response: Check carrier database file for invalid sequence.

ICA4110 **Carrier *carrier* is not valid and is being skipped.**

Explanation: Carrier cannot be used for paging purposes.

User Response: Check validity of carrier.

ICA4111 **Cannot add carrier to list.**

Explanation: Carrier cannot be added to list.

User Response: Check carrier validity and phone numbers.

ICA4112 Carrier name is missing or too long on line *lineNumber*.

Explanation: Carrier name is missing.

User Response: Add carrier using smitty/SMIT menu.

ICA4113 Cannot allocate new paging carrier: *carrier*.

Explanation: Carrier cannot be allocated to list.

User Response: Check carrier validity and phone numbers.

ICA4114 Value on line *lineNumber* is too long.

Explanation: Encountered a line that is too long in carrier database file.

User Response: Cleanup the long line in carrier database file.

ICA4115 Duplicate tag *tag* on line *lineNumber* ignored.

Explanation: Encountered a duplicate tag.

User Response: Remove the duplicate tag from carrier database file.

ICA4116 Value on line *lineNumber* does not exist.

Explanation: Encountered a blank field.

User Response: Use smitty/SMIT to add a value in blank field.

ICA4117 Value must be either Y, Yes, N or No on line *lineNumber*.

Explanation: This field requires either a Y, Yes, N or No.

User Response: Use smitty/SMIT to add or change valid data.

ICA4118 Value must be greater than 0 on line *lineNumber*.

Explanation: This field must be positive.

User Response: Change value using smitty/SMIT to a positive value.

ICA4119 Invalid value on line *lineNumber*.

Explanation: Encountered an invalid value on specified line.

User Response: Change value using smitty/SMIT menu.

ICA4120 Carrier *name* is not valid and is being skipped.

Explanation: Encountered an invalid carrier.

User Response: Add a valid carrier using smitty/SMIT menu.

ICA4121 Cannot add carrier to list.

Explanation: Cannot add carrier to the paging list.

User Response: Check carrier validity.

ICA4122 Duplicate tag *tag* on line *lineNumber* ignored.

Explanation: Encountered a duplicate tag in a carrier stanza.

User Response: Cleanup the carrier stanza containing duplicate values.

ICA4123 Error(*program*): Could not get IPC token: *errNo*

Explanation: Program could not get IPC token.

ICA4124 Error(*program*): Error *pageqErr* while reading queue.

Explanation: Program could not read queue.

ICA4125 *count* Queue entries.

Explanation: Informational message.

ICA4126 Message with ID *id* deleted.

Explanation: Informational message.

ICA4127 ID *id* not in queue.

Explanation: Informational message.

ICA4128 **Error(program): Error pageqErr while attempting to delete ID id.**

Explanation: Tried to deleted an ID of the queue.

ICA4129 **Key is: entryKey content is @ ptr: ptr.**

Explanation: Informational message only.

ICA4130 **Modem Characteristics:**

Explanation: Modem initialization information.

ICA4131 **Name: modemName**

Explanation: Modem initialization information.

ICA4132 **Init: initString**

Explanation: Modem initialization information.

ICA4133 **Command mode: command**

Explanation: Modem initialization information.

ICA4134 **Command terminator: 0xterminator**

Explanation: Modem initialization information.

ICA4135 **Dial: dial**

Explanation: Modem initialization information.

ICA4136 **Dial pause: pause**

Explanation: Modem initialization information.

ICA4137 **Dial #: diallb**

Explanation: Modem initialization information.

ICA4138 **Dial *: dialstar**

Explanation: Modem initialization information.

ICA4139 **Hangup: hangup**

Explanation: Modem initialization information.

ICA4140 **Valid command response: validCommandresp**

Explanation: Modem initialization information.

ICA4141 **Valid connect: validConnect**

Explanation: Modem initialization information.

ICA4142 **Echo: echo**

Explanation: Modem initialization information.

ICA4143 **Modem debug record: PUTS(id) txd-> outStr**

Explanation: Modem handshaking information.

ICA4144 **Modem debug record: PUTC(id) txd-> outStr**

Explanation: Modem handshaking information.

ICA4145 **Modem debug record: GET rxd-> record id**

Explanation: Modem handshaking information.

ICA4146 **Modem debug record: INPUT(record id**

Explanation: Modem handshaking information.

ICA4147 **Modem debug record:) rxd->**

Explanation: Modem handshaking information.

ICA4148 **Modem debug record: WAITFOR(record id**

Explanation: Modem handshaking information.

ICA4149 **Could not unblock child signal.**

Explanation: Unblocks the SIGCHLD signal.

ICA4150 **Could not block the child signal.**

Explanation: Blocks the SIGCHLD signal.

ICA4151 **Warm start file filePathname does not exist.**

Explanation: Informational message.

ICA4152 **Cannot open warm start file filePathname**

Explanation: Informational message.

ICA4153 **Line is too long in warm start file**
filePathname.

Explanation: The warm start file contains some invalid characters.

ICA4154 **Warm start file *filePathname* has data**
that is not being used.

Explanation: Informational message.

ICA4155 **Warm start file *filePathname* is empty.**

Explanation: Informational message.

ICA4156 **Line *lineNumber* of warm start file**
***filePathname* has bad addressee**
address, ignored.

Explanation: Warm start file has some invalid characters. Informational message.

ICA4157 **Line *lineNumber* of warm start file**
***filePathname* has bad format, ignored.**

Explanation: Warm start file has some invalid characters. Informational message.

ICA4158 **Line *lineNumber* of warm start file**
***filePathname* has no message, ignored.**

Explanation: Warm start file has no messages. Informational message.

ICA4159 **Error queueing line *lineNumber* of warm**
start file *filePathname*, ignored.

Explanation: Warm start file has some invalid characters. Informational message.

ICA4160 **Warm start of *count* messages from file**
***filePathname* complete.**

Explanation: Informational message.

ICA4161 **Error(*program*): Too many consecutive**
child errors.

Explanation: Too many child errors in a row. This occurs if either the carrier or the modem definition file has some invalid characters.

User Response: Check carrier database file and modem definition file using smitty/SMIT menu.

ICA4162 **Child cannot exec *program* : *errno*.**

Explanation: Possible system error.

ICA4163 **Error(*errno*): Child cannot fork child :**
***program name*.**

Explanation: Possible system error.

ICA4164 **Could not create paging carrier list.**

Explanation: Internal program error.

ICA4165 **Errors in paging carrier file *carrierFile***

Explanation: Carrier database contains some invalid data.

User Response: Check carrier database file using smitty/SMIT menu.

ICA4166 **Informational message. IPC key is:**
***0xIpcKey*.**

Explanation: Informational message.

ICA4167 **Could not create queue, page_q_err:**
***pageQerr*.**

Explanation: Failed while trying to create queue.

ICA4168 **Paging Warm Start file created at *time***

Explanation: Informational message.

ICA4169 **priority -p *priority numPager* from**
objfrom message

Explanation: Informational message.

ICA4170 **priority -p *priority alpaPager@carrier***
from *from message*

Explanation: Informational message.

ICA4171 **priority -p *priority -n numPager@carrier***
from *from message*

Explanation: Informational message.

ICA4172 **End of pager warm start file.**

Explanation: Informational message. Denotes end of message.

ICA4173 **Cannot write into warm start file**
warmstrtFile.

Explanation: Warm start file may not exist.

ICA4174 *time* **STATUS-REQUEST from** *user@host*

Explanation: Displays the status request information.

ICA4175 *time* **SUMMARY-REQUEST from**
user@host.

Explanation: Displays the summary request information.

ICA4176 *count* **queue entries.**

Explanation: Counts the number of queue entries in pager queue.

ICA4177 **Oldest entry: ID** *id* **received at** *time.*

Explanation: Displays the oldest entry in queue.

ICA4178 **Re-attaching memory after expansion**
failed.

Explanation: Possible system error.

ICA4179 **Re-attaching memory after expansion**
failed to align.

Explanation: Possible system error.

ICA4180 **Could not down PAGE_Q semaphore in**
page_q_print() : *errno.*

Explanation: Possible system error.

ICA4181 **Could not up PAGE_Q semaphore in**
page_q_print() : *errno.*

Explanation: Possible system error.

ICA4182 **link** *headLink* **-> message ID:** *id.*

Explanation: Informational message.

ICA4183 **Priority:** *priority.*

Explanation: Informational message.

ICA4184 **Person:** *name.*

Explanation: Informational message.

ICA4185 **Carrier:** *carrier.*

Explanation: Informational message.

ICA4186 **Mesg:** *message.*

Explanation: Informational message.

ICA4187 **Could not get shared RAM :** *errno.*

Explanation: Possible system error.

ICA4188 **Could not get attached shared RAM :**
errno.

Explanation: Possible system error.

ICA4189 **Could not get PAGE_Q semaphore.**

Explanation: Possible system error.

ICA4190 **Could not initialize PAGE_Q**
semaphore in page_q_create() : *errno.*

Explanation: Possible system error.

ICA4191 **Could not set PAGE_Q semaphore in**
page_q_create() : *errno.*

Explanation: Possible system error.

ICA4192 **Could not down PAGE_Q semaphore in**
page_q_empty() : *errno.*

Explanation: Possible system error.

ICA4193 **Could not up PAGE_Q semaphore in**
page_q_empty() : *errno.*

Explanation: Possible system error.

ICA4194 **Could not down PAGE_Q semaphore in**
page_q_enq(name,message) : *errno.*

Explanation: Possible system error.

ICA4195 Could not up PAGE_Q semaphore in page_q_enq() : *errno*.

Explanation: Possible system error.

ICA4196 page_q_enq(): ID(*id*) Pri(*priority*) Person(*name*) Mesg(*message*).

Explanation: Informational message.

ICA4197 Could not down PAGE_Q semaphore in page_q_head() : *errno*.

Explanation: Possible system error.

ICA4198 Could not up PAGE_Q semaphore in page_q_head() : *errno*.

Explanation: Possible system error.

ICA4199 Could not down PAGE_Q semaphore in page_q_first() : *errno*.

Explanation: Possible system error.

ICA4200 Could not up PAGE_Q semaphore in page_q_first() : *errno*.

Explanation: Possible system error.

ICA4201 Could not down PAGE_Q semaphore in page_q_next() : *errno*.

Explanation: Possible system error.

ICA4202 Could not up PAGE_Q semaphore in page_q_next() : *errno*.

Explanation: Possible system error.

ICA4203 Could not down PAGE_Q semaphore in page_q_tail() : *errno*.

Explanation: Possible system error.

ICA4204 Could not up PAGE_Q semaphore in page_q_tail() : *errno*.

Explanation: Possible system error.

ICA4205 Could not down PAGE_Q semaphore in page_q_del() : *errno*.

Explanation: Possible system error.

ICA4206 Could not up PAGE_Q semaphore in page_q_del() : *errno*.

Explanation: Possible system error.

ICA4207 page_q_del(*ID*).

Explanation: Debug information.

ICA4208 Could not down PAGE_Q semaphore in page_q_deq() : *errno*.

Explanation: Possible system error.

ICA4209 Could not up PAGE_Q semaphore in page_q_deq() : *errno*.

Explanation: Possible system error.

ICA4210 page_q_del(): ID(*id*) Pri(*priority*) Person(*name*) Mesg(*message*).

Explanation: Informational message.

ICA4211 Could not down PAGE_Q semaphore in page_q_walk() : *errno*.

Explanation: Possible system error.

ICA4212 Could not up PAGE_Q semaphore in page_q_walk() : *errno*.

Explanation: Possible system error.

ICA4213 PAGE_Q is full.

Explanation: The paging queue is full.

User Response: Send the page after some time.

ICA4300 Hanging up.

Explanation: Hanging up the call.

ICA4301 Initializing modem ..

Explanation: Initializing modem with the init string.

ICA4302 **Dialing**

Explanation: Dialing the phone number.

ICA4303 **Waiting for connection.**

Explanation: Waiting for the modem connection

ICA4304 **CONNECTED *speed***

Explanation: Connecting at the indicated speed (baud rate)

ICA4305 **CONNECTED!!!!!!!**

Explanation: Connected to the pager service provider

ICA4306 **Requesting prompt for Automatic Mode.**

Explanation: Requesting prompt for automatic mode. Waiting for "ID="

ICA4307 **Prompt OK.....**

Explanation: Got "ID=" back from the provider.

ICA4308 **Sending Automatic Mode Request.**

Explanation: Sending ID and SST over to the pager service provider

ICA4309 **Send Automatic Mode RequestOK!**

Explanation: Got [p back. Means communication successful

ICA4310 **Sending out message**

Explanation: Sending out message block over

ICA4311 **Waiting for result**

Explanation: Waiting for the confirmation

ICA4312 **Ack received. Page successful**

ICA4313 **Nak received, Resend block. Attempt *NakTries***

Explanation: Nak received. Pager provider is asking for resend

ICA4314 **Transaction error. Resend block. Attempt *RsTries***

Explanation: Transaction error. Resending the block over.

ICA4315 **Carrier Terminate Connection.**

Explanation: Pager provider terminated the conversation. Call provider for the problem

ICA4350 ***fwpage [carrier="..."] [modem="..."] [ID="..."] [msg="..."]***

Explanation: fwpage usage. Check your parameters and try again

ICA4351 *This file not exist*

Explanation: Check the file to see if it is under the right directory. The files carriers.cfg, modems.cfg, and pager.cfg must be created before using this code.

ICA4352 *What file corrupted*

Explanation: File has been modified by user and not in the stanza format. All attributes should be entered through GUI.

ICA4353 *What too long, please shorten it and try again*

Explanation: The 'What' parameter is too long. Shorten it and try again.

ICA4354 *What wrong.*

Explanation: If baud rate wrong, the valid options are: 600, 1200, 2400, 4800, 9600, 14400. If data bit per byte wrong, the valid options are: 7, 8. If stop bits wrong, the valid options are: 1,2. If out line prefix wrong, the inputs should only be numbers. If paging method wrong, only TAP is supported in this version. If pager ID error, check to see if it is all numbers. If parity wrong, the valid options are: O(odd), E(even), N(none), S(space), M(mark). If COM port wrong, the valid options are: COM1, COM2 COM port should be less than 10 in this version. If message character wrong, check the message to see if there is special character in it.

ICA4355 Set Parameters in *where* error.

Explanation: Unable to set parameters in [where]. Check parameters and try again.

ICA4356 when *When*, COM port reading error.

Explanation: COM port reading error. Set modem echo on and try again

ICA4357 when *Where*, COM port writing error.

Explanation: COM port write error.

ICA4358 Set *What* error

Explanation: Pager code is reporting the error indicated by 'What'. Check the log file to pin down the error.

ICA4359 Max tries exceed in *Where*. Abort program

Explanation: Tried to open com port 60 times in 60 minutes. All failed. If this is the case, check the hardware connection. Or, tried to send pager message 10 times in 10 minutes. All failed. If this is the case, the pager provider might be down.

ICA4360 Unknown character in Carrier phone number: **pCarrierPhoneNum*

Explanation: an unrecognized character found in the carrier phone number. Please check the number and try again.

ICA4361 Warning!!! Paging provider's modem normally should be less than 2400.

Explanation: This is just a warning. Paging provider's modem speed is normally set less than 2400.

ICA4362 Unable to initialize modem

Explanation: Change modem initialization string and try again.

ICA4363 Modem returned Error.

Explanation: Modem communication error

ICA4364 *tries* try on open Com port error. Retry in 1 minute

Explanation: Open com port error. Probably another program is using it. Automatically retry in 1 minutes

ICA4365 Send page failed on *tries* try. Retry in 1 minutes

Explanation: Send page failed. Check log file to find out the exact reason.

ICA4366 Message too long, truncated

Explanation: Just a warning. Message length is too long. Truncate to fit in.

ICA4367 Reset Max message length to the internal defined value:*msg-length*

Explanation: Reset the max message length to the default value, because user defined message length is larger than the internal defined, which is 80.

ICA4368 Action: *Where* error

Explanation: If opening COM port error, check configuration and try again. If close COM handle error, system problem. If purge COM error, system problem. If send dial command error, dialing command problem. Check to see if it is a Hayes compatible modem. If send ID request error, check if the pager provider supports TAP protocol and phone number is correct for TAP service. If send automatic prompt error, check if the pager service works correctly. If send message error, check the log file to pin down the cause of failure. If prompt error, unable to get a prompt back from the pager provider.

ICA4369 Too many transaction error. aborting

Explanation: Too many transaction errors, abort this try.

ICA4370 Too many Nak received, aborting the program

Explanation: Too many Nak received from the page provider, abort this try.

ICA4371 *szComPort* on COM port with function *FunctionName* return *Error Number*

Explanation: check the parameters and try again.

ICA4372 **Modem return error message.....**
ReturnMessage

Explanation: Messages are: Not connected. Ringing, but not connected. No carrier. No dial tone. Busy. No answer.

ICA4373 (*function name*) **Unknown response code from modem or carrier:** *char1*, *char2*.

Explanation: This message reports a response from the modem or carrier, that the Firewall's paging feature does not recognize. *char1* and *char2* are the ascii (hex) codes for the 1st 2 characters in the response.

User Response: Use this information when consulting your modem instructions or your carrier to determine the meaning of the unknown response.

ICA5005 **SKIT initialization failed. Return code is:** *return code*

Explanation: Secure socket initialization failed, return code from SKIT dispalyed.

ICA5014 **Remote Client Tunnel Server listening port** *server port #*

Explanation: Port number configured for *sslrctd* is displayed.

ICA5015 **Accepted connection from** *chp0.chp1.chp2.chp3*

Explanation: Client's IP address is displayed.

ICA5017 **Unable to get secure socket. Function** *skit_secure_soc_init* **retcode is:***function retcode*

Explanation: Cannot get secure socket because *skit_secure_soc_init()* failed.

ICA5018 **The slave server cipher specs used are** *spec1 spec2 spec3*

Explanation: Cihper specifications are displayed.

ICA5019 **Cannot get Free Homenet IP pool.**

Explanation: Dynamic filters problem.

ICA5020 **Cannot open remote client config file.**

Explanation: File */etc/security/rcsfile.cfg* is unavailable.

User Response: Check file presence and it's contents.

ICA5021 **Cannot find 'keyword' keyword.**

Explanation: File */etc/security/rcsfile.cfg* doesn't have this keyword.

User Response: Check and correct */etc/security/rcsfile.cfg*.

ICA5024 **Function** *skit_secure_soc_write()* **error in** *routine name*.

Explanation: *skit_secure_soc_write()* failed in this routine.

ICA5025 **Function** *skit_secure_soc_write()* **error in** *ACKClient()*.

Explanation: *skit_secure_soc_write()* failed in *ACKClient()* routine.

ICA5026 **Invalid return code received from Client in** *routine name*.

Explanation: Unexpected return code received from client in this routine.

ICA5027 **Received return code for wrong request from Client in** *routine name*.

Explanation: Request code in return code message is unexpected in this routine.

ICA5028 **Invalid Login Request.**

Explanation: Format of login request message is invalid.

ICA5030 **Unknown Remote Client ID :** *remote client ID*

Explanation: This user ID is unknown for firewall machine.

User Response: Correct user's information for this remote client.

ICA5031 **Function skit_secure_soc_write error in RCTLoginPhase.**

Explanation: skit_secure_soc_write() failed for login phase.

ICA5035 **Invalid Logout Request.**

Explanation: Format of logout request message is invalid.

ICA5067 **Invalid packet received.**

Explanation: Received packet format is invalid.

ICA5078 **Get unrecognized request in SvrReqHandler()**

Explanation: Unrecognized request received and will be ignored.

ICA5082 **Tunnel to client *remote client ID* has been disconnected.**

Explanation: Tunnel for the remote client with this ID was disconnected.

ICA5086 **ID: *userid* not defined.**

Explanation: This user ID does not exist on firewall machine.

ICA5087 **Authentication failed for '*userid*'.**

Explanation: Authentication failed for this user ID.

ICA5089 **Function rcFilterClear() failed. Return code is *return code*.**

Explanation: rcFilterClear() failed with this return code.

User Response: Check IPSEC LAN client presence. These products can't coexist.

ICA5090 **Function rcFilterInit() failed. Return code is *return code***

Explanation: rcFilterInit() failed with this return code.

ICA5091 **Function TunnelUp() cannot run executable file *command line*.**

Explanation: Displayed command line failed system() call.

ICA5092 **Cannot get keyring password from recoverstash function call.**

Explanation: Cannot recover keyring password from the stash file.

ICA8001 **SYSLOG/udp: unknown service**

ICA8002 *function_name* function failed - *errno*, **errno2 = 0x*errno2***

Explanation: Processing terminates because syslogd could not perform the specified function. The errno information is appended to the error message.

User Response: Contact the system programmer. SystemProgrammer :Use the errno information to determine the cause of the failure.

ICA8004 **Error detected on AF_INET socket, \slogd will no longer monitor socket**

ICA8006 **Unknown priority name \'*priority*\'**

Explanation: A priority name found in the configuration file is not valid.

User Response: Contact the system programmer. SystemProgrammer :Check the configuration file.

ICA8007 **Unknown facility name \'*facility*\'**

Explanation: A facility name found in the configuration file is not valid.

User Response: Contact the system programmer. SystemProgrammer :Check the configuration file.

ICA8008 **Message from SYSLOG@*hostname* at *timestamp* ...**

Explanation: The syslog daemon configuration file contained an entry to send syslog messages to all logged on users. This message will be sent to all users who are currently logged on to the system where the syslog daemon is running.

User Response: None SystemProgrammer :None

ICA8009 **SYSLOGD exiting on signal** *signal*

Explanation: The syslog daemon received a signal that has caused the syslog daemon to exit.

User Response: None SystemProgrammer :None

ICA8010 **SYSLOGD restarted**

ICA8012 **SYSLOGD unable to record to SMF -**
error_text

Explanation: An error occurred while writing a record to SMF. The error text information is appended to the error message.

User Response: Contact the system programmer. SystemProgrammer :Use the error text information to determine the cause of the SMF write failure.

ICA8013 **Update process status failed, return**
code = 0xreturn_code

Explanation: An error occurred while attempting to update the status of the syslogd process for the Firewall kernel process. The return code outlines the specific error that was returned from the update process status call.

User Response: Contact the system programmer. SystemProgrammer :Contact the service representative.

ICA8014 **Unknown option (-startup_option)**
specified on SYSLOGD invocation

Explanation: An error occurred while attempting to start the syslogd daemon process. The option specified is not supported on the invocation of syslogd.

User Response: Check the startup options and restart the syslogd daemon. SystemProgrammer :If the problem persists, contact the service representative.

ICA8015 **Configuration file entry (config_data) is**
not valid

Explanation: An error occurred while attempting to parse a configuration entry from the SYSLOG configuration file.

User Response: Check the configuration file entries and restart the syslogd daemon. SystemProgrammer :If the problem persists, contact the service representative.

ICA8016 **function_name failed for filename - errno**

Explanation: An error occurred while attempting to perform the specified function for the specified device. The errno information is appended to the error message.

User Response: Verify that the specified device exists and retry the request. If the problem persists, contact the system programmer. SystemProgrammer :If the problem persists, contact the service representative.

ICA8050 **function failed. error_text**

Explanation: An error was encountered executing the function displayed in the message. Additional information about the error is given by the error text.

User Response: Correct the error specified in the message and, if necessary, retry the operation.

ICA8051 **function failed: return code =**
0xreturn_code

Explanation: An error was encountered executing the function displayed in the message. The return code from the specified function is also displayed.

User Response: Correct the error specified in the message and, if necessary, retry the operation.

ICA8052 **FWSTACKD activating filter logging for**
stack_name.

Explanation: FWSTACKD is attempting to activate packet filter logging.

System Action: The program continues.

ICA8053 **FWSTACKD cannot activate filter**
logging for stack_name. error_text

Explanation: Activation of packet filter logging failed for the reason described in the accompanying error message.

System Action: Filter logging will not be performed.

User Response: Use the error message to correct the error, then reactivate filters logging with **fwfilter cmd=startlog**.

ICA8054 **FWSTACKD activating NAT logging for**
stack_name.

Explanation: FWSTACKD is attempting to activate network address translation (NAT) logging.

System Action: The program continues.

ICA8055 **FWSTACKD cannot activate NAT logging for *stack_name*. *error_text***

Explanation: Activation of network address translation (NAT) logging failed for the reason described in the accompanying error message.

System Action: Network address translation logging will not be performed.

User Response: Use the error message to correct the error, then reactivate network address translation logging with **fwnat cmd=startlog**.

ICA8056 **FWSTACKD activating NAT for *stack_name*.**

Explanation: FWSTACKD is attempting to activate network address translation (NAT).

System Action: The program continues.

ICA8057 **FWSTACKD cannot activate NAT for *stack_name*. *error_text***

Explanation: Activation of network address translation (NAT) failed for the reason described in the accompanying error message.

System Action: Network address translation will not be performed.

User Response: Use the error message to correct the error, then reactivate network address translation with **fwnat cmd=update**.

ICA8058 **FWSTACKD reactivating tunnel definitions for *stack_name*.**

Explanation: FWSTACKD is attempting to reactivate tunnel definitions that were active when the system was stopped.

System Action: The program continues.

ICA8059 **FWSTACKD cannot reactivate tunnel definitions for *stack_name*. *error_text***

Explanation: Activation of tunnel definitions failed for the reason described in the accompanying error message.

System Action: Tunnels definitions are not activated.

User Response: Use the error message to correct the error, then reactivate tunnel definitions with **fwtunnl cmd=activate**.

ICA8060 **FWSTACKD activating filter and Socks rules for *stack_name*.**

Explanation: FWSTACKD is attempting to activate the current set of packet filter rules and Socks rules.

System Action: The program continues.

ICA8061 **FWSTACKD cannot activate filter and Socks rules for *stack_name*. *error_text***

Explanation: Activation of filter rules and Socks rules failed for the reason described in the accompanying error message.

System Action: Default filter rules will be in effect. Local access will be permitted, and all other access will be denied.

User Response: Use the error message to correct the error, then reactivate filters and Socks rules with **fwfilter cmd=update**.

ICA8062 **FWSTACKD activating RealAudio support for *stack_name*.**

Explanation: FWSTACKD is attempting to activate RealAudio support.

System Action: The program continues.

ICA8063 **FWSTACKD cannot activate RealAudio support for *stack_name*. *error_text***

Explanation: Activation of RealAudio support failed for the reason described in the accompanying error message.

System Action: RealAudio services are unavailable.

User Response: Use the error message to identify the error, then fix the error and activate RealAudio with **fwaudio cmd=change**.

ICA8064 ***function failed*. *error_text***

Explanation: An error was encountered executing the function displayed in the message. Additional information about the error is given by the error text.

User Response: Correct the error specified in the message and, if necessary, retry the operation.

ICA9000 **IBM Firewall evaluation expires in number of days.**

Explanation: This software is branded as an evaluation copy and will disable itself as indicated.

ICA9001 **File System Integrity Checker Warning**
- *warning description text*

Explanation: fwfschk found a discrepancy in the filesystem - potential threat

ICA9002 **last message repeated %1\$d times**

Explanation: Message generated by AIX syslogd when an identical message is logged without any intervening message. The message is kept here for Log Monitor to be able to detect the condition. This message must be in whatever language the real syslogd message is being written.

ICA9003 **Authentication failed for user *name* on the configuration server.**

Explanation: FW configuration server is unable to authenticate the indicated user.

User Response: See your FW administrator.

ICA9004 **User *name* successfully authenticated on the configuration server.**

Explanation: FW configuration server authenticated the indicated user.

ICA9005 **Starting remote configuration server.**

Explanation: Configuration server has been started.

ICA9006 **Ending remote configuration server.**

Explanation: Configuration server is ending.

ICA9007 **Remote configuration server unable to open message catalog.**

Explanation: One or more message catalogs used by the remote configuration server may be missing.

User Response: See your FW administrator.

ICA9008 **Remote configuration server failed on getpeername(): error *errno*.**

Explanation: Configuration server is unable to obtain information about the client.

User Response: See your FW administrator.

ICA9009 **Remote configuration server failed on getsockname(): error *errno*.**

Explanation: Configuration server is unable to obtain information about itself.

User Response: See your FW administrator.

ICA9010 **Remote configuration server failed obtaining adapter information.**

Explanation: Configuration server is unable to obtain adapter information.

User Response: See your FW administrator.

ICA9011 **Configuration server not enabled for remote configuration.**

Explanation: Configuration server has local=yes set in its configuration file and the client is on a remote machine.

User Response: See your FW administrator.

ICA9012 **Remote configuration server unable to read logon request.**

Explanation: Configuration server cannot read in the client logon request.

User Response: See your FW administrator.

ICA9013 **Remote configuration server received incorrect logon request.**

Explanation: Logon request contained incorrect information.

User Response: See your FW administrator.

ICA9014 **Remote configuration server unable to create pipe.**

Explanation: Configuration server cannot create a pipe for authentication.

User Response: See your FW administrator.

ICA9015 **Remote configuration server unable to create process.**

Explanation: Configuration server cannot create a process for authentication.

User Response: See your FW administrator.

ICA9016 **Starting EFM daemon.**

Explanation: The EFM daemon has been started on the managed firewall.

ICA9017 **Ending EFM daemon; rc = *value*.**

Explanation: The EFM daemon is ending with the specified return code.

ICA9018 **EFM daemon unable to open message catalog.**

Explanation: One or more message catalogs used by the EFM daemon may be missing.

User Response: See your FW administrator.

ICA9020 **Unable to switch the running user ID.**

Explanation: failed to make the system call to switch the running user ID.

User Response: See your FW administrator.

ICA9021 **This firewall does not support *logon* mode.**

Explanation: This firewall does not support this particular mode.

User Response: See your FW administrator.

ICA9022 ***user* is not authorized to logon to the firewall in *logon* mode.**

Explanation: This username is not authorized to logon using this particular mode.

User Response: See your FW administrator.

ICA9023 **Unable to load EFM DLL.**

Explanation: failed to load the efm dll.

User Response: See your FW administrator.

ICA9024 **Transfer request started by *user* to firewall *machine*.**

Explanation: The transfer operation has started.

ICA9025 **Transfer request ended with return code *return code*.**

Explanation: The transfer operation has completed.

ICA9026 **Transfer request received from *user* on firewall *machine* on *time*.**

Explanation: The transfer operation has started at the specified time.

ICA9027 **File *filename* in function *function* added to transfer request.**

Explanation: The file specified is going to be transferred.

ICA9028 **Activate request started by *user* to firewall *machine*.**

Explanation: The activate operation has started.

ICA9029 **Activate request ended with return code *return code*.**

Explanation: The activate operation has completed.

ICA9030 **Activate request received from *user* on firewall *machine* on *time*.**

Explanation: The activate operation has started at the specified time.

ICA9031 **Activate of function *function* ended with return code *return code*.**

Explanation: Activation of the specified function has completed.

ICA9032 **NAT configuration updated at *time* on *date*.**

Explanation: NAT configuration has been updated.

ICA9033 **NAT support (*level version.release*) initialized at *time* on *date*.**

Explanation: Firewall NAT support has been initialized.

ICA9034 NAT support deactivated at *time on date*.

Explanation: NAT support has been disabled.

ICA9035 NAT unable to allocate Registered Address for Secured Address *Secured IP Address*.

Explanation: Secured Address not translated because there are no available addresses in the Registered Address pool.

ICA9036 NAT released Registered Address *Registered IP Address to address pool*.

Explanation: Registered Address has been released to registered IP address pool.

ICA9037 Firewall interfaces being updated automatically on *time_and_date*.

Explanation: The Firewall initialization program has called **UpdateInterfaces()** to trigger the automatic update of the Firewall interfaces file, *fwadpt.cfg*.

System Action: none

User Response: none

ICA9038 Interface *address* has been removed from Firewall configurations.

Explanation: The dotted-decimal address listed had been listed in the Firewall config file *fwadpt.cfg*, but was not known to the TCP stack, and has therefore been removed from the config file.

System Action: none

User Response: none

ICA9039 Interface *address* has been added to the Firewall configuration.

Explanation: The dotted-decimal address listed was found by the TCP stack but had not been found in the Firewall config file *fwadpt.cfg*, and has therefore been added to the config file.

System Action: none

User Response: none

ICA9040 Interface *address* mask was updated from *oldmask* to *newmask*.

Explanation: The mask in the *fwadpt.cfg* file did not match what was found installed on the hardware. The correct mask field was updated in the *fwadpt.cfg* file.

System Action: none

User Response: none

ICA9041 No interfaces were found on this machine.

Explanation: No adapter interfaces were found on this machine.

System Action: none

User Response: none

ICA9042 NAT activated with a working many-to-one address *many-to-one address*.

Explanation: NAT has successfully initialized and now is active. If the address is 0, this implies that many-to-one translation is inactive.

System Action: none

User Response: none

ICA9043 NAT failed to initialize with returned code *rc*.

Explanation: NAT failed to initialize and is inactive.

System Action: No NAT function will be invoked.

User Response: If user wants NAT functionality, look at the returned code and make adjustment to correct it. If problem persists, contact IBM service.

ICA9044 NAT deactivated.

Explanation: NAT has successfully deactivated and is now inactive.

System Action: none

User Response: none

ICA9045 NAT allocated address:port *address:port* for secured address:port *secured address:port*

Explanation: NAT has allocated the address:port from the address pool on behalf of the secured host.

System Action: none

User Response: none

ICA9046 NAT is unable to allocate many-to-one address for secured address *secured address*

Explanation: NAT has run out of ports with the many-to-one address.

System Action: The local host's packet has been dropped.

User Response: This implies that there are too many outstanding connections. An administrator might want to decrease the time-out associated with the many-to-one address in an attempt to eliminate idle translation table entries more quickly.

ICA9047 NAT deallocated address:port *address:port* from secured address:port *secured address:port*.

Explanation: NAT returned the specified address:port pair to the available pool.

System Action: none

User Response: none

ICA9048 NAT detected a fragmented packet with protocol:*protocol* address:port *address:port* secured address:port *secured address:port*.

Explanation: NAT has detected either a fragment FTP control packet or a fragmented ICMP error message. NAT will translate a fragmented FTP control packet, however the payload is not examined. If this was a fragmented PORT command, the FTP data will fail because the IP address contained in the message is not translated. If the packet is a fragmented ICMP error message, it will be dropped.

System Action: See explanation.

User Response: If this happens repeatedly, notify IBM service.

ICA9049 NAT detected an out of order fragment from *source address* to *destination address* that could not be translated.

Explanation: NAT has detected a fragmented datagram that has arrived prior to the first fragment of the datagram.

System Action: NAT cannot translate the fragment correctly and the datagram is dropped.

User Response: If this happens repeatedly, notify IBM service.

ICA9050 NAT failed to translate a packet with protocol:*protocol*, source address:port *address:port*, destination address:port *secured address:port*, with returned code *rc*.

Explanation: NAT failed to translate a packet.

System Action: packet is dropped.

User Response: If this happens repeatedly, notify IBM service.

ICA9051 NAT detected a packet arrived with protocol:*protocol* to address:port *address:port* from secured address:port *secured address:port*

Explanation: NAT has detected the arrival of a packet.

System Action: none

User Response: none

ICA9052 NAT detected a packet leaving with protocol:*protocol* to address:port *address:port* from secured address:port *secured address:port*

Explanation: NAT has detected the departure of a packet.

System Action: none

User Response: none

ICA9053 *stringValue filename* in %3\$d

Explanation: debugging

System Action: none

User Response: none

ICA9054 IP address:*address* cannot be used as a many-to-one address and a nonsecure/secure interface address simultaneously.

Explanation: They cannot be identical.

System Action: The requested action is not performed.

User Response: Choose a different nonsecure/secure address or different many-to-one address.

ICA9055 **NAT detected an out of order fragment from *source address* to *destination address* that could be translated.**

Explanation: NAT has detected an internal or final datagram fragment that has arrived out of order.

System Action: NAT was able to translate the fragment correctly and so did not drop the datagram.

User Response: none

ICA9060 **Fatal configuration server initialization error - *socket()*: *system error message***

Explanation: Configuration server initialization failed, daemon terminated.

User Response: Correct the indicated system problem and restart the configuration server.

ICA9061 **Fatal configuration server initialization error - *listen()*: *system error message***

Explanation: Configuration server initialization failed, daemon terminated.

User Response: Correct the indicated system problem and restart the configuration server.

ICA9062 **Fatal configuration server error - *main accept()*: *system error message***

Explanation: Configuration server main routine failed, daemon terminated.

User Response: Correct the indicated system problem and restart the configuration server.

ICA9063 **Configuration server error - *failing function*: *return code = 0xfunction return code***

Explanation: The configuration server detected an error in the indicated function. The daemon terminates.

User Response: Correct the indicated system problem and restart the configuration server.

ICA9064 **Unknown option -*value* ignored.**

Explanation: Indicated option was specified and is not recognized.

ICA9065 **Configuration server error - *failing function*: *system error message***

Explanation: The configuration server detected an error in the indicated function. The daemon terminates.

User Response: Correct the indicated system problem and restart the configuration server.

ICA9066 **Insufficient memory: configuration server: *malloc(bytes)* returned NULL in function *function_name*.**

Explanation: Unable to allocate enough memory - malloc returned NULL.

ICA9067 **Bind failed, address: *port* already in use.**

Explanation: Port address given is currently in use.

System Action: The configuration server terminates.

User Response: Connect to the Configuration Server using a different port address, or contact your Firewall administrator.

ICA9068 **-*value* option failed or was specified incorrectly.**

Explanation: The indicated option failed or was specified incorrectly.

System Action: The configuration server terminates.

User Response: Correct the usage of the indicated option and restart the configuration server.

ICA9069 **SSL Initialization failed.**

Explanation: The SSL encryption environment was unable to be initialized or the handshake with the partner failed.

System Action: The configuration server terminates.

User Response: See your Firewall administrator to verify the SSL environment.

Appendix B. Hardening for the AIX System Configuration

Hardening is a process that maximizes security and efficiency by turning off unnecessary daemons and disabling unauthorized user IDs. Hardening is part of installation of the IBM Firewall software and edits the system resources that might compromise security.

The hardening process:

- Removes these daemons from `/etc/rc.tcpip`: `lpd`, `routed`, `gated`, `portmap`, `timed`, `snmpd`, `rwhod`, `fs`, `sendmail`, `named` and `dpid2`.
- Disables the AIX Common Desktop Environment.
- Removes all unnecessary programs from `inittab`. Everything is taken out of `/etc/inittab` EXCEPT `init`, `brc`, `powerfail`, `rc`, `fbcheck`, `srcmstr`, `rctcpip`, `cron`, `cons`, `logsymp`, `diagd`, `acfgd`, `pmd` and `tty`.
- Disables all logins for users except `root`, `daemon`, `bin`, `adm`, `nobody`, and any previous IBM Firewall users.
- Sets owners to `root` for all files and directories that have no owners, and sets permissions to zero.
- During the hardening process, `root` is converted to a firewall user with remote logins disabled. Also, any previous firewall users using a downlevel version are migrated to the new version.
- Disables nonsecure applications by setting permissions to zero. These nonsecure applications are: `tftp`, `utftp`, `tftpd`, `uucpd`, `rcp`, `rlogin`, `rlogind`, `rsh`, and `rshd`.
- Disables everything in `/etc/inetd.conf` EXCEPT: `ftp`, `telnet`, and `ibmfwracs` (remote config server daemon).

When the hardening process is complete, the file system integrity checker database is generated.

Appendix C. SNMP Management Information Base (MIB)

This appendix gives detail of the Firewall MIB.

```
-- FW fwMib Definitions
--
IBMFW-fwMib DEFINITIONS ::= BEGIN

-- This component represents a system configured with IBM's
-- Internet Connection IBM Firewall (FW) product.
-- The groups defined are as follows:

-- the FW Syslog Trap group
-- the FW Server Status Trap group
-- the FW Component ID group
-- the FW Software Component Information group
-- the FW Subagent group
-- the FW Server Table group
-- the ftpd Proxy Server Group
-- the telnetd Proxy Server Group
-- the Mail Server Table Group
-- the Log File Management Table group
-- the FW Server Status Table group
-- the FW Server Concurrency Status Table group
-- the FW Configuration File Table group
-- the FW Filter Status group
-- the Network Configuration group
-- the Threshold Configuration Table group
-- the Active IP Tunnel Table group
-- the Network Address Translation Table

IMPORTS
    Counter, enterprises
        FROM RFC1155-SMI
    OBJECT-TYPE
        FROM RFC-1212
    DisplayString
        FROM RFC1213-fwMib
    TRAP-TYPE
        FROM RFC-1215;

--
-- The MIB was registered under the original name Secured Network Gateway
-- (SNG).
--

internet          OBJECT IDENTIFIER ::= { iso org(3) dod(6) 1 }
private           OBJECT IDENTIFIER ::= { internet 4}
enterprises       OBJECT IDENTIFIER ::= { private 1 }
ibm               OBJECT IDENTIFIER ::= { enterprises 2 }
ibmProd           OBJECT IDENTIFIER ::= { ibm 6 }
ibmSNG            OBJECT IDENTIFIER ::= { ibmProd 129 }
fwMib             OBJECT IDENTIFIER ::= { ibmSNG 1 }
fwSubagent        OBJECT IDENTIFIER ::= { ibmSNG 2 }

-- FW Syslog Trap Group =====
fwSyslogTrapGrp OBJECT IDENTIFIER ::= {fwMib 1}
```

```

fwSyslogFacility OBJECT-TYPE
    SYNTAX          DisplayString (SIZE (0..20))
    ACCESS          not-accessible
    STATUS          mandatory
    DESCRIPTION    "syslog facility that generated the record."
    --              The string can be one of the following:
    --              "local1"
    --              "local4"
    ::= {fwSyslogTrapGrp 1}

fwSyslogLogFileName OBJECT-TYPE
    SYNTAX          DisplayString (SIZE (0..255))
    ACCESS          not-accessible
    STATUS          mandatory
    DESCRIPTION    "File where the syslog record was entered."
    ::= {fwSyslogTrapGrp 2}

fwSyslogDate OBJECT-TYPE
    SYNTAX          DisplayString (SIZE (0..50))
    ACCESS          not-accessible
    STATUS          mandatory
    DESCRIPTION    "Date of the syslog record."
    ::= {fwSyslogTrapGrp 3}

fwSyslogTime OBJECT-TYPE
    SYNTAX          DisplayString (SIZE (0..20))
    ACCESS          not-accessible
    STATUS          mandatory
    DESCRIPTION    "Time of the syslog record."
    ::= {fwSyslogTrapGrp 4}

fwSyslogHost OBJECT-TYPE
    SYNTAX          DisplayString (SIZE (0..255))
    ACCESS          not-accessible
    STATUS          mandatory
    DESCRIPTION    "Host in the syslog record."
    ::= {fwSyslogTrapGrp 5}

fwSyslogPid OBJECT-TYPE
    SYNTAX          INTEGER
    ACCESS          not-accessible
    STATUS          mandatory
    DESCRIPTION    "Process id in the syslog record."
    ::= {fwSyslogTrapGrp 6}

fwSyslogMsgText OBJECT-TYPE
    SYNTAX          DisplayString (SIZE (0..255))
    ACCESS          not-accessible
    STATUS          mandatory
    DESCRIPTION    "Message text in the syslog record."
    ::= {fwSyslogTrapGrp 7}

-- FW Server Status Trap Group =====
fwSvrStatTrapGrp OBJECT IDENTIFIER ::= {fwMib 2}

fwSvrName OBJECT-TYPE
    SYNTAX          DisplayString (SIZE (0..255))
    ACCESS          not-accessible
    STATUS          mandatory
    DESCRIPTION    "The server's name."

```



```

        ::= {fwSvrStatTrapGrp 1}

fwSvrProgram OBJECT-TYPE
    SYNTAX      DisplayString (SIZE (0..255))
    ACCESS      not-accessible
    STATUS      mandatory
    DESCRIPTION "The server executable name"
    ::= {fwSvrStatTrapGrp 2}

fwSvrState OBJECT-TYPE
    SYNTAX      DisplayString (SIZE (0..50))
    ACCESS      not-accessible
    STATUS      mandatory
    DESCRIPTION "The server's current running state."
    --          The string can be one of the following:
    --          "running"
    --          "not running"
    ::= {fwSvrStatTrapGrp 3}

fwSvrStateValue OBJECT-TYPE
    SYNTAX      INTEGER
    {
        vUnknown      (0),
        vNotRunning   (1),
        vRunning      (2)
    }
    ACCESS      not-accessible
    STATUS      mandatory
    DESCRIPTION "The server's current running state (integer form)."
    ::= {fwSvrStatTrapGrp 4}

fwSvrTrapTimestamp OBJECT-TYPE
    SYNTAX      DisplayString (SIZE (0..30))
    ACCESS      not-accessible
    STATUS      mandatory
    DESCRIPTION "Timestamp at which the server status trap generated."
    ::= {fwSvrStatTrapGrp 5}

fwSvrTrapHost OBJECT-TYPE
    SYNTAX      DisplayString (SIZE (0..255))
    ACCESS      not-accessible
    STATUS      mandatory
    DESCRIPTION "Hostname from where the trap generated."
    ::= {fwSvrStatTrapGrp 6}

-- FW Component ID Group =====
fwComponentIdGroup OBJECT IDENTIFIER ::= {fwMib 3}

fwManufacturer OBJECT-TYPE
    SYNTAX      DisplayString (SIZE (0..32))
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "The company that produced this component."
    --          The string is: "IBM Corporation".
    ::= {fwComponentIdGroup 1}

fwProduct OBJECT-TYPE
    SYNTAX      DisplayString (SIZE (0..255))
    ACCESS      read-only
    STATUS      mandatory

```

```

DESCRIPTION      "The name of this component or product."
--              The string is: "IBM FW SNMP Subagent."
::= {fwComponentIdGroup 2}

fwVersion OBJECT-TYPE
SYNTAX           DisplayString (SIZE (0..16))
ACCESS           read-only
STATUS           mandatory
DESCRIPTION      "The version string for this component."
::= {fwComponentIdGroup 3}

fwVerify OBJECT-TYPE
SYNTAX           INTEGER
-- {
--   vAnErrorOccurred;CheckStatusCode           (0),
--   vThisComponentDoesNotExist                 (1),
--   vTheVerifyIsNotSupported                   (2),
--   vReserved                                   (3),
--   vComponent'sFunctionalityUntested          (4),
--   vComponent'sFunctionalityUnknown           (5),
--   vComponentIsNotFunctioningCorrectly        (6),
--   vComponentFunctionsCorrectly               (7)
-- }
ACCESS           read-only
STATUS           mandatory
DESCRIPTION      "A code that provides a level of verification
                  that the component is still installed and
                  working. This value is 2 for this release."
::= {fwComponentIdGroup 4}

fwVerifyString OBJECT-TYPE
SYNTAX           DisplayString (SIZE (0..32))
ACCESS           read-only
STATUS           mandatory
DESCRIPTION      "A string that corresponds to the aVerify
                  value. The string for this release will
                  be: Verify is not supported."
::= {fwComponentIdGroup 5}

-- FW Software Component Information Group =====
fwSoftwareCompInfoGroup OBJECT IDENTIFIER ::= {fwMib 4}

fwMajorVersion OBJECT-TYPE
SYNTAX           DisplayString (SIZE (0..16))
ACCESS           read-only
STATUS           mandatory
DESCRIPTION      "Major version of this fwSoftware component."
::= {fwSoftwareCompInfoGroup 1}

fwMinorVersion OBJECT-TYPE
SYNTAX           DisplayString (SIZE (0..16))
ACCESS           read-only
STATUS           mandatory
DESCRIPTION      "Minor version of this fwSoftware component."
::= {fwSoftwareCompInfoGroup 2}

fwRevision OBJECT-TYPE
SYNTAX           DisplayString (SIZE (0..16))
ACCESS           read-only
STATUS           mandatory
DESCRIPTION      "Revision of this fwSoftware component."

```

```

        ::= {fwSoftwareCompInfoGroup 3}

fwTargetOperatingSystem OBJECT-TYPE
    SYNTAX          INTEGER
    ACCESS          read-only
    STATUS          mandatory
    DESCRIPTION    "The operating system for which this fwSoftware
                    component is intended."
    ::= {fwSoftwareCompInfoGroup 4}

fwLanguageEdition OBJECT-TYPE
    SYNTAX          DisplayString (SIZE (0..16))
    ACCESS          read-only
    STATUS          mandatory
    DESCRIPTION    "The language edition of this fwSoftware
                    component. This string will be : English."
    ::= {fwSoftwareCompInfoGroup 5}

fwTargetOsString OBJECT-TYPE
    SYNTAX          DisplayString (SIZE (0..32))
    ACCESS          read-only
    STATUS          mandatory
    DESCRIPTION    "The operating system for which this fwSoftware
                    component is intended. This is AIX for this release."
    ::= {fwSoftwareCompInfoGroup 6}

-- FW Subagent Group =====
fwSubagentGroup OBJECT IDENTIFIER ::= {fwMib 5}

fwSubagtName OBJECT-TYPE
    SYNTAX          DisplayString (SIZE (0..32))
    ACCESS          read-only
    STATUS          mandatory
    DESCRIPTION    "The name of this subagent is IBM FW
                    Subagent. The string is: IBM FW Subagent."
    ::= {fwSubagentGroup 1}

fwSubagtUpTime OBJECT-TYPE
    SYNTAX          DisplayString (SIZE (0..26))
    ACCESS          read-only
    STATUS          mandatory
    DESCRIPTION    "The date and time the FW subagent was last
                    started."
    ::= {fwSubagentGroup 2}

fwCritlogPoll OBJECT-TYPE
    SYNTAX          INTEGER
    ACCESS          read-write
    STATUS          mandatory
    DESCRIPTION    "Polling interval (in minutes) for critlog thread."
    ::= {fwSubagentGroup 3}

fwCritlogTimestamp OBJECT-TYPE
    SYNTAX          DisplayString (SIZE (0..26))
    ACCESS          read-write
    STATUS          mandatory
    DESCRIPTION    "Beginning timestamp for monitoring critlog records."
    ::= {fwSubagentGroup 4}

fwCritlogLocation OBJECT-TYPE
    SYNTAX          DisplayString (SIZE (0..255))

```

```

ACCESS          read-write
STATUS          mandatory
DESCRIPTION     "Location of critlog file(s)."  

::= {fwSubagentGroup 5}

fwSvrStatPoll OBJECT-TYPE
SYNTAX          INTEGER
ACCESS          read-write
STATUS          mandatory
DESCRIPTION     "Polling interval (in minutes) for server status thread."  

::= {fwSubagentGroup 6}

-- FW Server Table Group =====
-- FwSvrEntry has to start with an upper case otherwise mosy gives an error

fwSvrTbl OBJECT-TYPE
SYNTAX          SEQUENCE OF FwSvrEntry
ACCESS          not-accessible
STATUS          mandatory
DESCRIPTION     "A list of entries for FW servers configured on this  

host." ::= {fwMib 6}

aFwSvrEntry OBJECT-TYPE
SYNTAX          FwSvrEntry
ACCESS          not-accessible
STATUS          mandatory
DESCRIPTION     ""
INDEX          {fwServerName}
::= {fwSvrTbl 1}

FwSvrEntry ::= SEQUENCE
{
    fwServerName          DisplayString,
    fwServerSocketType   DisplayString,
    fwServerProtocol     DisplayString,
    fwServerWait         DisplayString,
    fwServerUser         DisplayString,
    fwServerProgram      DisplayString,
    fwServerArgs         DisplayString }

fwServerName OBJECT-TYPE
SYNTAX          DisplayString (SIZE (0..50))
ACCESS          read-only
STATUS          mandatory
DESCRIPTION     "The name of the FW or inet server."  

-- The string can be one of the following:  

-- "Unknown"  

-- "FTPD Proxy"  

-- "Telnetd Proxy"  

-- "Http Proxy"  

-- ... or any service in the file /etc/services.  

::= {aFwSvrEntry 1}

fwServerSocketType OBJECT-TYPE
SYNTAX          DisplayString (SIZE (0..50))
ACCESS          read-only
STATUS          mandatory
DESCRIPTION     "The type of socket the server is using."  

-- The string can be one of the following:  

-- "stream"  

-- "dgram"  

-- "sunrpc_udp"

```

```

--                                     "sunrpc_tcp"
 ::= {aFwSvrEntry 2}

fwServerProtocol OBJECT-TYPE
SYNTAX          DisplayString (SIZE (0..50))
ACCESS          read-only
STATUS          mandatory
DESCRIPTION    "The communication protocol the server is using."
--             The string can be one of the protocols found in the
--             file /etc/protocols.
 ::= {aFwSvrEntry 3}

fwServerWait OBJECT-TYPE
SYNTAX          DisplayString (SIZE (0..50))
ACCESS          read-only
STATUS          mandatory
DESCRIPTION    "The wait/no wait attribute of the server."
--             The string can be one of the following:
--             "wait"
--             "nowait"
 ::= {aFwSvrEntry 4}

fwServerUser OBJECT-TYPE
SYNTAX          DisplayString (SIZE (0..50))
ACCESS          read-only
STATUS          mandatory
DESCRIPTION    "The username inetd uses to start the server."
 ::= {aFwSvrEntry 5}

fwServerProgram OBJECT-TYPE
SYNTAX          DisplayString (SIZE (0..255))
ACCESS          read-only
STATUS          mandatory
DESCRIPTION    "Full pathname of the server that inetd should execute."
 ::= {aFwSvrEntry 6}

fwServerArgs OBJECT-TYPE
SYNTAX          DisplayString (SIZE (0..50))
ACCESS          read-only
STATUS          mandatory
DESCRIPTION    "Command line arguments used in starting the server."
 ::= {aFwSvrEntry 7}

-- ftpd Proxy Server Group =====
fwFtpdSvrGrp OBJECT IDENTIFIER ::= {fwMib 7}

fwFtpdSvrName OBJECT-TYPE
SYNTAX          DisplayString (SIZE (0..50))
ACCESS          read-only
STATUS          mandatory
DESCRIPTION    "The name of the FW server."
--             The string is "pftpd"
 ::= {fwFtpdSvrGrp 1}

fwFtpdSvrSocketType OBJECT-TYPE
SYNTAX          DisplayString (SIZE (0..50))
ACCESS          read-only
STATUS          mandatory
DESCRIPTION    "The type of socket the server is using."
--             The string can be one of the following:

```

```

--                "stream"
--                "dgram"
--                "sunrpc_udp"
--                "sunrpc_tcp"
::= {fwFtpdSvrGrp 2}

fwFtpdSvrProtocol OBJECT-TYPE
SYNTAX            DisplayString (SIZE (0..50))
ACCESS            read-only
STATUS            mandatory
DESCRIPTION       "The communication protocol the server is using."
--               The string can be one of the protocols found in the
--               file /etc/protocols.
::= {fwFtpdSvrGrp 3}

fwFtpdSvrWait OBJECT-TYPE
SYNTAX            DisplayString (SIZE (0..50))
ACCESS            read-only
STATUS            mandatory
DESCRIPTION       "The wait/no wait attribute of the server."
--               The string can be one of the following:
--               "wait"
--               "nowait"
::= {fwFtpdSvrGrp 4}

fwFtpdSvrUser OBJECT-TYPE
SYNTAX            DisplayString (SIZE (0..50))
ACCESS            read-only
STATUS            mandatory
DESCRIPTION       "The user who invoked the server."
::= {fwFtpdSvrGrp 5}

fwFtpdSvrProgram OBJECT-TYPE
SYNTAX            DisplayString (SIZE (0..255))
ACCESS            read-only
STATUS            mandatory
DESCRIPTION       "Full pathname of the server that inetd should execute."
::= {fwFtpdSvrGrp 6}

fwFtpdSvrArgs OBJECT-TYPE
SYNTAX            DisplayString (SIZE (0..50))
ACCESS            read-only
STATUS            mandatory
DESCRIPTION       "Command line arguments used in starting the server."
::= {fwFtpdSvrGrp 7}

-- telnetd Proxy Server Group =====
fwTelnetdSvrGrp OBJECT IDENTIFIER ::= {fwMib 8}

fwTelnetdSvrName OBJECT-TYPE
SYNTAX            DisplayString (SIZE (0..50))
ACCESS            read-only
STATUS            mandatory
DESCRIPTION       "The name of the FW server."
--               The string is "ptelnetd"
::= {fwTelnetdSvrGrp 1}

fwTelnetdSvrSocketType OBJECT-TYPE
SYNTAX            DisplayString (SIZE (0..50))
ACCESS            read-only
STATUS            mandatory

```

```

DESCRIPTION      "The type of socket the server is using."
--              The string can be one of the following:
--              "stream"
--              "dgram"
--              "sunrpc_udp"
--              "sunrpc_tcp"
::= {fwTelnetdSvrGrp 2}

fwTelnetdSvrProtocol OBJECT-TYPE
SYNTAX          DisplayString (SIZE (0..50))
ACCESS          read-only
STATUS          mandatory
DESCRIPTION     "The communication protocol the server is using."
--              The string can be one of the protocols found in the
--              file /etc/protocols.
::= {fwTelnetdSvrGrp 3}

fwTelnetdSvrWait OBJECT-TYPE
SYNTAX          DisplayString (SIZE (0..50))
ACCESS          read-only
STATUS          mandatory
DESCRIPTION     "The wait/no wait attribute of the server."
--              The string can be one of the following:
--              "wait"
--              "nowait"
::= {fwTelnetdSvrGrp 4}

fwTelnetdSvrUser OBJECT-TYPE
SYNTAX          DisplayString (SIZE (0..50))
ACCESS          read-only
STATUS          mandatory
DESCRIPTION     "The user who invoked the server."
::= {fwTelnetdSvrGrp 5}

fwTelnetdSvrProgram OBJECT-TYPE
SYNTAX          DisplayString (SIZE (0..255))
ACCESS          read-only
STATUS          mandatory
DESCRIPTION     "Full pathname of the server that inetd should execute."
::= {fwTelnetdSvrGrp 6}

fwTelnetdSvrArgs OBJECT-TYPE
SYNTAX          DisplayString (SIZE (0..50))
ACCESS          read-only
STATUS          mandatory
DESCRIPTION     "Command line arguments used in starting the server."
::= {fwTelnetdSvrGrp 7}

-- FW Mail Servers Group =====
- FwMailSvrEntry has to start with an upper case otherwise mosy gives an error

fwMailSvrTbl OBJECT-TYPE
SYNTAX          SEQUENCE OF FwMailSvrEntry
ACCESS          not-accessible
STATUS          mandatory
DESCRIPTION     "A list of entries for FW mail servers configured
on this host." ::= {fwMib 9}

aFwMailSvrEntry OBJECT-TYPE
SYNTAX          FwMailSvrEntry
ACCESS          not-accessible

```

```

STATUS          mandatory
DESCRIPTION     ""
INDEX           {fwMailSecDomName}
::= {fwMailSvrTbl 1}

FwMailSvrEntry ::= SEQUENCE
{
    fwMailSecDomName          DisplayString,
    fwMailSecNKSVr           DisplayString,
    fwMailPubDomName          DisplayString
}

fwMailSecDomName OBJECT-TYPE
SYNTAX          DisplayString (SIZE (0..255))
ACCESS          read-only
STATUS          mandatory
DESCRIPTION     "The name of the FW Secure Domain Name."
--             The first column in /etc/security/mail.conf
::= {aFwMailSvrEntry 1}

fwMailSecNKSVr OBJECT-TYPE
SYNTAX          DisplayString (SIZE (0..255))
ACCESS          read-only
STATUS          mandatory
DESCRIPTION     "The name of the FW Secure Network Mail Server."
--             The second column in /etc/security/mail.conf
::= {aFwMailSvrEntry 2}

fwMailPubDomName OBJECT-TYPE
SYNTAX          DisplayString (SIZE (0..255))
ACCESS          read-only
STATUS          mandatory
DESCRIPTION     "The name of the FW Public Domain Name."
--             The second column in /etc/security/mail.conf
::= {aFwMailSvrEntry 3}

-- Log File Management Table Group =====
fwLogFileMgmtTbl OBJECT-TYPE
SYNTAX          SEQUENCE OF FwLogFileMgmtEntry
ACCESS          not-accessible
STATUS          mandatory
DESCRIPTION     "table of log files to be Managed"
::= {fwMib 10}

aFwLogFileMgmtEntry OBJECT-TYPE
SYNTAX          FwLogFileMgmtEntry
ACCESS          not-accessible
STATUS          mandatory
DESCRIPTION     ""
INDEX           {fwLogFileName}
::= {fwLogFileMgmtTbl 1}

FwLogFileMgmtEntry ::= SEQUENCE
{
    fwLogFileName          DisplayString,
    fwLogDaysInLog         INTEGER,
    fwLogArchive           DisplayString,
    fwLogDaysInArc         INTEGER,
    fwLogWorkSpace         DisplayString,
    fwLogComments          DisplayString
}

```



```

}

fwLogFileName OBJECT-TYPE
    SYNTAX      DisplayString (SIZE (0..255))
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "Name of the log file to be Managed."
    ::= {aFwLogFileMgmtEntry 1}

fwLogDaysInLog OBJECT-TYPE
    SYNTAX      INTEGER
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "Days to keep in logfile."
    ::= {aFwLogFileMgmtEntry 2}

fwLogArchive OBJECT-TYPE
    SYNTAX      DisplayString (SIZE (0..255))
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "Archive name."
    ::= {aFwLogFileMgmtEntry 3}

fwLogDaysInArc OBJECT-TYPE
    SYNTAX      INTEGER
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "Days to keep in archive."
    ::= {aFwLogFileMgmtEntry 4}

fwLogWorkSpace OBJECT-TYPE
    SYNTAX      DisplayString (SIZE (0..255))
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "directory where log Management operations
take place."
    ::= {aFwLogFileMgmtEntry 5}

fwLogComments OBJECT-TYPE
    SYNTAX      DisplayString (SIZE (0..255))
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "comments."
    ::= {aFwLogFileMgmtEntry 6}

-- FW Server Status Table Group =====
fwSvrStatTbl OBJECT-TYPE
    SYNTAX      SEQUENCE OF FwSvrStatEntry
    ACCESS      not-accessible
    STATUS      mandatory
    DESCRIPTION "A list of status entries for FW servers configured on
this host."
    ::= {fwMib 11}

aFwSvrStatEntry OBJECT-TYPE
    SYNTAX      FwSvrStatEntry
    ACCESS      not-accessible
    STATUS      mandatory
    DESCRIPTION ""
    INDEX      {fwSvrStatServerName}
    ::= {fwSvrStatTbl 1}

```

```

FwSvrStatEntry ::= SEQUENCE
{
    fwSvrStatServerName      DisplayString,
    fwSvrStatServerState    DisplayString
}

fwSvrStatServerName OBJECT-TYPE
SYNTAX      DisplayString (SIZE (0..50))
ACCESS      read-only
STATUS      mandatory
DESCRIPTION "The name of the FW or inet server."
--          The string can be one of the following:
--          "fwsubagt"
--          "inetd"
--          "fwpagerd"
--          "fwmaild"
--          "named"
 ::= {aFwSvrStatEntry 1}

fwSvrStatServerState OBJECT-TYPE
SYNTAX      DisplayString (SIZE (0..50))
ACCESS      read-only
STATUS      mandatory
DESCRIPTION "Is the server running?"
--          The string can be one of the following:
--          "unknown"
--          "running"
--          "not running"
 ::= {aFwSvrStatEntry 2}

-- FW Server Concurrency Status Table Group =====
fwSvrConStatTbl OBJECT-TYPE
SYNTAX      SEQUENCE OF FwSvrConStatEntry
ACCESS      not-accessible
STATUS      mandatory
DESCRIPTION ""
 ::= {fwMib 12}

aFwSvrConStatEntry OBJECT-TYPE
SYNTAX      FwSvrConStatEntry
ACCESS      not-accessible
STATUS      mandatory
DESCRIPTION ""
INDEX       {fwSvrConStatServerName}
 ::= {fwSvrConStatTbl 1}

FwSvrConStatEntry ::= SEQUENCE
{
    fwSvrConStatServerName      DisplayString,
    fwSvrConStatSessions        INTEGER
}

fwSvrConStatServerName OBJECT-TYPE
SYNTAX      DisplayString (SIZE (0..50))
ACCESS      read-only
STATUS      mandatory
DESCRIPTION "The name of the FW."
--          The string can be one of the following:
--          "FTPD Proxy"
--          "Telnetd Proxy"

```

```

--                               "SOCKS Server"
--                               "Http Proxy"
 ::= {aFwSvrConStatEntry 1}

fwSvrConStatServerSessions OBJECT-TYPE
SYNTAX          INTEGER
ACCESS          read-only
STATUS          mandatory
DESCRIPTION     "Number of concurrent sessions."
 ::= {aFwSvrConStatEntry 2}

-- FW Configuration File Table Group =====
fwCfgFileTbl OBJECT-TYPE
SYNTAX          SEQUENCE OF FwCfgFileEntry
ACCESS          not-accessible
STATUS          mandatory
DESCRIPTION     "Information about FW and FW-related configuration files."
 ::= {fwMib 13}

aFwCfgFileEntry OBJECT-TYPE
SYNTAX          FwCfgFileEntry
ACCESS          not-accessible
STATUS          mandatory
DESCRIPTION     ""
INDEX          {fwCfgFileName}
 ::= {fwCfgFileTbl 1}

FwCfgFileEntry ::= SEQUENCE
{
    fwCfgFileName      DisplayString,
    fwCfgUser          DisplayString,
    fwCfgGroup         DisplayString,
    fwCfgTimeStamp     DisplayString,
    fwCfgSize          INTEGER,
    fwCfgStatus        INTEGER,
    fwChecksum         INTEGER
}

fwCfgFileName OBJECT-TYPE
SYNTAX          DisplayString (SIZE (0..255))
ACCESS          read-only
STATUS          mandatory
DESCRIPTION     "The monitored file."
 ::= {aFwCfgFileEntry 1}

fwCfgUser OBJECT-TYPE
SYNTAX          DisplayString (SIZE (0..50))
ACCESS          read-only
STATUS          mandatory
DESCRIPTION     "The user who owns the file."
 ::= {aFwCfgFileEntry 2}

fwCfgGroup OBJECT-TYPE
SYNTAX          DisplayString (SIZE (0..50))
ACCESS          read-only
STATUS          mandatory
DESCRIPTION     "The file's primary group."
 ::= {aFwCfgFileEntry 3}

```

```

fwCfgTimeStamp OBJECT-TYPE
    SYNTAX          DisplayString (SIZE (0..50))
    ACCESS          read-only
    STATUS          mandatory
    DESCRIPTION    "Current timestamp."
    ::= {aFwCfgFileEntry 4}

fwCfgSize OBJECT-TYPE
    SYNTAX          INTEGER
    ACCESS          read-only
    STATUS          mandatory
    DESCRIPTION    "File size in bytes."
    ::= {aFwCfgFileEntry 5}

fwCfgStatus OBJECT-TYPE
    SYNTAX          INTEGER
    {
        vUnknown      (0),
        vNotFound     (1),
        vFound        (2)
    }
    ACCESS          read-only
    STATUS          mandatory
    DESCRIPTION    "Is the file found?"
    ::= {aFwCfgFileEntry 6}

fwChecksum OBJECT-TYPE
    SYNTAX          INTEGER
    ACCESS          read-only
    STATUS          mandatory
    DESCRIPTION    "checksum on the file"
    ::= {aFwCfgFileEntry 7}

-- FW Filter Status Group =====
fwFilterStatGrp OBJECT IDENTIFIER
::= {fwMib 14}

fwFilterNumIfs OBJECT-TYPE
    SYNTAX          INTEGER
    ACCESS          read-only
    STATUS          mandatory
    DESCRIPTION    "Number of secure interfaces defined."
    ::= {fwFilterStatGrp 1}

fwFilterNumRules OBJECT-TYPE
    SYNTAX          INTEGER
    ACCESS          read-only
    STATUS          mandatory
    DESCRIPTION    "Number of rules in filter list."
    ::= {fwFilterStatGrp 2}

fwFilterLevel OBJECT-TYPE
    SYNTAX          DisplayString (SIZE (0..50))
    ACCESS          read-only
    STATUS          mandatory
    DESCRIPTION    "Netinet filter support code level."
    ::= {fwFilterStatGrp 3}

-- XXX Can status be anything other than 'not available'?

```

```

fwFilterStat OBJECT-TYPE
    SYNTAX          DisplayString (SIZE (0..50))
    ACCESS          read-only
    STATUS          mandatory
    DESCRIPTION    "Status of filter support code."
    ::= {fwFilterStatGrp 4}

-- XXX Can status be anything other than 'not available'?
fwPktLogStat OBJECT-TYPE
    SYNTAX          DisplayString (SIZE (0..50))
    ACCESS          read-only
    STATUS          mandatory
    DESCRIPTION    "Status of packet logging."
    ::= {fwFilterStatGrp 5}

fwFilterRulesTimeStamp OBJECT-TYPE
    SYNTAX          DisplayString (SIZE (0..100))
    ACCESS          read-only
    STATUS          mandatory
    DESCRIPTION    "Timestamp of last update to rules."
    ::= {fwFilterStatGrp 6}

fwFilterNumRulesUpdates OBJECT-TYPE
    SYNTAX          INTEGER
    ACCESS          read-only
    STATUS          mandatory
    DESCRIPTION    "Number of updates to rules since initialization."
    ::= {fwFilterStatGrp 7}

-- Network Configuration Group =====
fwNetCfgGrp OBJECT IDENTIFIER
::= {fwMib 15}

fwSecDomName OBJECT-TYPE
    SYNTAX          DisplayString (SIZE (0..100))
    ACCESS          read-only
    STATUS          mandatory
    DESCRIPTION    "Secure domain name."
    ::= {fwNetCfgGrp 1}

fwNonSecDomSvrTbl OBJECT-TYPE
    SYNTAX          SEQUENCE OF FwNonsecDomSvrEntry
    ACCESS          not-accessible
    STATUS          mandatory
    DESCRIPTION    ""
    ::= {fwNetCfgGrp 2}

aFwNonsecDomSvrEntry OBJECT-TYPE
    SYNTAX          FwNonsecDomSvrEntry
    ACCESS          not-accessible
    STATUS          mandatory
    DESCRIPTION    ""
    INDEX          {fwNonSecSvraddr}
    ::= {fwNonSecDomSvrTbl 1}

FwNonsecDomSvrEntry ::= SEQUENCE
{
    fwNonSecSvrAddr          DisplayString
}

```

```

fwNonSecSvrAddr OBJECT-TYPE
    SYNTAX          DisplayString (SIZE (0..100))
    ACCESS          read-only
    STATUS          mandatory
    DESCRIPTION    ""
    ::= {aFwNonsecDomSvrEntry 1}

fwSecDomSvrTbl OBJECT-TYPE
    SYNTAX          SEQUENCE OF FwSecDomSvrEntry
    ACCESS          not-accessible
    STATUS          mandatory
    DESCRIPTION    ""
    ::= {fwNetCfgGrp 3}

aFwSecDomSvrEntry OBJECT-TYPE
    SYNTAX          FwSecDomSvrEntry
    ACCESS          not-accessible
    STATUS          mandatory
    DESCRIPTION    ""
    INDEX          {fwSecSvrAddr}
    ::= {fwSecDomSvrTbl 1}

FwSecDomSvrEntry ::= SEQUENCE
{
    fwSecSvrAddr          DisplayString
}

fwSecSvrAddr OBJECT-TYPE
    SYNTAX          DisplayString (SIZE (0..100))
    ACCESS          read-only
    STATUS          mandatory
    DESCRIPTION    ""
    ::= {aFwSecDomSvrEntry 1}

-- Threshold Configuration Group =====
fwThrCfgGrp OBJECT IDENTIFIER ::= {fwMib 16}

fwMailToTbl OBJECT-TYPE
    SYNTAX          SEQUENCE OF FwMailToEntry
    ACCESS          not-accessible
    STATUS          mandatory
    DESCRIPTION    "Table of users to notify of threshold violations"
    ::= {fwThrCfgGrp 1}

aFwMailToEntry OBJECT-TYPE
    SYNTAX          FwMsgThrEntry
    ACCESS          not-accessible
    STATUS          mandatory
    DESCRIPTION    ""
    INDEX          {fwMailToId}
    ::= {fwMailToTbl 1}

FwMailToEntry ::= SEQUENCE
{
    fwMailToId          DisplayString,
    fwMailToComments   DisplayString
}

fwMailToId OBJECT-TYPE
    SYNTAX          DisplayString (SIZE (0..255))

```

```

ACCESS          read-only
STATUS          mandatory
DESCRIPTION     "mail address to send threshold violation notice to"
 ::= {aFwMailToEntry 1}

fwMailToComments OBJECT-TYPE
SYNTAX          DisplayString (SIZE (0..255))
ACCESS          read-only
STATUS          mandatory
DESCRIPTION     "comments"
 ::= {aFwMailToEntry 2}

fwCommand OBJECT-TYPE
SYNTAX          DisplayString (SIZE (0..255))
ACCESS          read-only
STATUS          mandatory
DESCRIPTION     "program executed when threshold is reached."
 ::= {fwThrCfgGrp 2}

fwCommandComments OBJECT-TYPE
SYNTAX          DisplayString (SIZE (0..255))
ACCESS          read-only
STATUS          mandatory
DESCRIPTION     "comments"
 ::= {fwThrCfgGrp 3}

fwSnglAuthThrCount OBJECT-TYPE
SYNTAX          INTEGER
ACCESS          read-only
STATUS          mandatory
DESCRIPTION     "number of failed authentication messages to be detected."
 ::= {fwThrCfgGrp 4}

fwSnglAuthThrTime OBJECT-TYPE
SYNTAX          INTEGER
ACCESS          read-only
STATUS          mandatory
DESCRIPTION     "Number of minutes to detect failed auth messages."
 ::= {fwThrCfgGrp 5}

fwSnglAuthPagerAlert OBJECT-TYPE
SYNTAX          DisplayString (SIZE (0..20))
ACCESS          read-only
STATUS          mandatory
DESCRIPTION     "Pager notification"
 ::= {fwThrCfgGrp 6}

fwSnglAuthThrComments OBJECT-TYPE
SYNTAX          DisplayString (SIZE (0..255))
ACCESS          read-only
STATUS          mandatory
DESCRIPTION     "comments"
 ::= {fwThrCfgGrp 7}

fwMultAuthThrCount OBJECT-TYPE
SYNTAX          INTEGER
ACCESS          read-only
STATUS          mandatory
DESCRIPTION     "number of failed authentication messages to be detected."
 ::= {fwThrCfgGrp 8}

```

```

fwMultAuthThrTime OBJECT-TYPE
    SYNTAX          INTEGER
    ACCESS          read-only
    STATUS          mandatory
    DESCRIPTION     "Number of minutes to detect failed auth messages."
    ::= {fwThrCfgGrp 9}

fwMultAuthPagerAlert OBJECT-TYPE
    SYNTAX          DisplayString (SIZE (0..20))
    ACCESS          read-only
    STATUS          mandatory
    DESCRIPTION     "Pager notification"
    ::= {fwThrCfgGrp 10}

fwMultAuthThrComments OBJECT-TYPE
    SYNTAX          DisplayString (SIZE (0..255))
    ACCESS          read-only
    STATUS          mandatory
    DESCRIPTION     "comments"
    ::= {fwThrCfgGrp 11}

fwHostAuthThrCount OBJECT-TYPE
    SYNTAX          INTEGER
    ACCESS          read-only
    STATUS          mandatory
    DESCRIPTION     "number of failed authentication messages to be detected."
    ::= {fwThrCfgGrp 12}

fwHostAuthThrTime OBJECT-TYPE
    SYNTAX          INTEGER
    ACCESS          read-only
    STATUS          mandatory
    DESCRIPTION     "Number of minutes to detect failed auth messages."
    ::= {fwThrCfgGrp 13}

fwHostAuthPagerAlert OBJECT-TYPE
    SYNTAX          DisplayString (SIZE (0..20))
    ACCESS          read-only
    STATUS          mandatory
    DESCRIPTION     "Pager notification"
    ::= {fwThrCfgGrp 14}

fwHostAuthThrComments OBJECT-TYPE
    SYNTAX          DisplayString (SIZE (0..255))
    ACCESS          read-only
    STATUS          mandatory
    DESCRIPTION     "comments"
    ::= {fwThrCfgGrp 15}

fwMsgThrTbl OBJECT-TYPE
    SYNTAX          SEQUENCE OF FwMsgThrEntry
    ACCESS          not-accessible
    STATUS          mandatory
    DESCRIPTION     "Table of message threshold definition entries"
    ::= {fwThrCfgGrp 16}

aFwMsgThrEntry OBJECT-TYPE
    SYNTAX          FwMsgThrEntry
    ACCESS          not-accessible
    STATUS          mandatory

```



```

DESCRIPTION      ""
INDEX            {fwMsgThrTag}
::= {fwMsgThrTbl 1}

FwMsgThrEntry ::= SEQUENCE
{
    fwMsgThrTag      DisplayString,
    fwMsgThrCount    INTEGER,
    fwMsgThrTime     INTEGER,
    fwMsgThrPagerAlert DisplayString,
    fwMsgThrComments DisplayString
}

fwMsgThrTag OBJECT-TYPE
SYNTAX      DisplayString (SIZE (0..20))
ACCESS      read-only
STATUS      mandatory
DESCRIPTION ""
::= {aFwMsgThrEntry 1}

fwMsgThrCount OBJECT-TYPE
SYNTAX      INTEGER
ACCESS      read-only
STATUS      mandatory
DESCRIPTION "number of failed authentication messages to be detected."
::= {aFwMsgThrEntry 2}

fwMsgThrTime OBJECT-TYPE
SYNTAX      INTEGER
ACCESS      read-only
STATUS      mandatory
DESCRIPTION "Number of minutes to detect failed auth messages."
::= {aFwMsgThrEntry 3}

fwMsgThrPagerAlert OBJECT-TYPE
SYNTAX      DisplayString (SIZE (0..20))
ACCESS      read-only
STATUS      mandatory
DESCRIPTION "Pager notification."
::= {aFwMsgThrEntry 4}

fwMsgThrComments OBJECT-TYPE
SYNTAX      DisplayString (SIZE (0..255))
ACCESS      read-only
STATUS      mandatory
DESCRIPTION ""
::= {aFwMsgThrEntry 5}

-- FW Active IP Tunnel Table Group =====
fwActiveTunnelGrp OBJECT IDENTIFIER ::= {fwMib 17}

fwIbmTunnelTbl OBJECT-TYPE
SYNTAX      SEQUENCE OF FwIbmTunnelEntry
ACCESS      not-accessible
STATUS      mandatory
DESCRIPTION "List of all IBM Tunnels "
::= {fwActiveTunnelGrp 1}

```

```

aFwIbmTunnelEntry OBJECT-TYPE
    SYNTAX          FwIbmTunnelEntry
    ACCESS          not-accessible
    STATUS          mandatory
    DESCRIPTION    "A list of all IBM Tunnels."
    INDEX          {fwIbmTunnelId}
    ::= {fwIbmTunnelTb1 1}

FwIbmTunnelEntry ::= SEQUENCE
{
    fwIbmTunnelId      INTEGER,
    fwIbmSrcAddr       DisplayString,
    fwIbmDestAddr      DisplayString,
    fwIbmEncription    DisplayString,
    fwIbmPolicy        DisplayString,
    fwIbmSessionLife   INTEGER,
    fwIbmInitFlag      DisplayString
}

fwIbmTunnelId OBJECT-TYPE
    SYNTAX          INTEGER
    ACCESS          read-only
    STATUS          mandatory
    DESCRIPTION    "The Identification number of the IBM Tunnel."
    ::= {aFwIbmTunnelEntry 1}

fwIbmSrcAddr OBJECT-TYPE
    SYNTAX          DisplayString (SIZE (0..100))
    ACCESS          read-only
    STATUS          mandatory
    DESCRIPTION    "The IP address of the local firewall."
    ::= {aFwIbmTunnelEntry 2}

fwIbmDestAddr OBJECT-TYPE
    SYNTAX          DisplayString (SIZE (0..100))
    ACCESS          read-only
    STATUS          mandatory
    DESCRIPTION    "The IP address of the partner firewall."
    ::= {aFwIbmTunnelEntry 3}

fwIbmEncryption OBJECT-TYPE
    SYNTAX          DisplayString (SIZE (0..20))
    ACCESS          read-only
    STATUS          mandatory
    DESCRIPTION    "Algorithm used for IP Packet encryption ."
    --             possible values are DES_CBC_8, CDMF, DES_CBC_4
    ::= {aFwIbmTunnelEntry 4}

fwIbmPolicy OBJECT-TYPE
    SYNTAX          DisplayString (SIZE (0..20))
    ACCESS          read-only
    STATUS          mandatory
    DESCRIPTION    "combination of encryption and authentication values."
    --             Possible values are encr/auth, auth/encr, encr only,
    --             auth only, none
    ::= {aFwIbmTunnelEntry 5}

fwIbmSessionLife OBJECT-TYPE
    SYNTAX          INTEGER
    ACCESS          read-only
    STATUS          mandatory

```

```

DESCRIPTION      "Time in minutes current session can be used."
--              Max time is 1440.
::= {aFwIbmTunnelEntry 6}

fwIbmInitFlag OBJECT-TYPE
SYNTAX          DisplayString (SIZE (0..20))
ACCESS          read-only
STATUS          mandatory
DESCRIPTION     "Identifies which partner starts the session
                negotiation."
--              Possible values are yes, no
::= {aFwIbmTunnelEntry 7}

fwManTunnelTbl OBJECT-TYPE
SYNTAX          SEQUENCE OF FwManTunnelEntry
ACCESS          not-accessible
STATUS          mandatory
DESCRIPTION     ""
::= {fwActiveTunnelGrp 2}

aFwManTunnelEntry OBJECT-TYPE
SYNTAX          FwManTunnelEntry
ACCESS          not-accessible
STATUS          mandatory
DESCRIPTION     "A list of all MAN Tunnels."
INDEX          {fwManTunnelId}
::= {fwManTunnelTbl 1}

FwManTunnelEntry ::= SEQUENCE
{
    fwManTunnelId      INTEGER,
    fwManSrcAddr       DisplayString,
    fwManDestAddr      DisplayString,
    fwManEncryption    DisplayString,
    fwManPolicy        DisplayString,
    fwManSessionLife   INTEGER,
    fwManTargetSPI     INTEGER
}

fwManTunnelId OBJECT-TYPE
SYNTAX          INTEGER
ACCESS          read-only
STATUS          mandatory
DESCRIPTION     "The Identification number of the Man Tunnel."
::= {aFwManTunnelEntry 1}

fwManSrcAddr OBJECT-TYPE
SYNTAX          DisplayString (SIZE (0..100))
ACCESS          read-only
STATUS          mandatory
DESCRIPTION     "The IP address of the local firewall."
::= {aFwManTunnelEntry 2}

fwManDestAddr OBJECT-TYPE
SYNTAX          DisplayString (SIZE (0..100))
ACCESS          read-only
STATUS          mandatory
DESCRIPTION     "The IP address of the partner firewall."
::= {aFwManTunnelEntry 3}

```

```

fwManEncryption OBJECT-TYPE
    SYNTAX          DisplayString (SIZE (0..20))
    ACCESS          read-only
    STATUS          mandatory
    DESCRIPTION    "Algorithm used for IP Packet encryption ."
    --             possible values are DES_CBC_8, CDMF, DES_CBC_4
    ::= {aFwManTunnelEntry 4}

fwManPolicy OBJECT-TYPE
    SYNTAX          DisplayString (SIZE (0..20))
    ACCESS          read-only
    STATUS          mandatory
    DESCRIPTION    "combination of encryption and authentication values."
    --             Possible values are encr/auth, auth/encr, encr only,
    --             auth only, none
    ::= {aFwManTunnelEntry 5}

fwManSessionLife OBJECT-TYPE
    SYNTAX          INTEGER
    ACCESS          read-only
    STATUS          mandatory
    DESCRIPTION    "Time in minutes manual tunnel will be operational."
    --             Max time is 44640.
    ::= {aFwManTunnelEntry 6}

fwManTargetSpi OBJECT-TYPE
    SYNTAX          INTEGER
    ACCESS          read-only
    STATUS          mandatory
    DESCRIPTION    "Target Security Parameter Index for manual tunnel."
    --             Valid values are 1- 9999
    ::= {aFwManTunnelEntry 7}

-- FW Network Address Translation Group =====
fwNatAddrTransGrp OBJECT IDENTIFIER
::= {fwMib 18}

fwNatReservedTbl OBJECT-TYPE
    SYNTAX          SEQUENCE OF FwResvRegisterEntry
    ACCESS          not-accessible
    STATUS          mandatory
    DESCRIPTION    ""
    ::= {fwNatAddrTransGrp 1}

aFwResvRegisterEntry OBJECT-TYPE
    SYNTAX          FwResvRegisterEntry
    ACCESS          not-accessible
    STATUS          mandatory
    DESCRIPTION    ""
    INDEX          {fwRegisteredIpAddr}
    ::= {fwNatReservedTbl 1}

FwResvRegisterEntry ::= SEQUENCE
{
    fwRegisteredIpAddr      DisplayString,
    fwRegisteredIpAddrMask DisplayString,
    fwNatTimeout           INTEGER
}

```

```

fwRegisteredIpAddr OBJECT-TYPE
    SYNTAX      DisplayString (SIZE (0..100))
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "Defines the IP addresses for outbound connections"
    ::= {aFwResvRegisterEntry 1}

```

```

fwRegisteredIpMask OBJECT-TYPE
    SYNTAX      DisplayString (SIZE (0..100))
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "The mask specifies the bits in the registered IP
    addr used to add a range of IP addr to the registered addr pool."
    ::= {aFwResvRegisterEntry 2}

```

```

fwNatTimeout OBJECT-TYPE
    SYNTAX      INTEGER
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "minutes an address translation can remain idle."
    ::= {aFwResvRegisterEntry 3}

```

```

fwNatTranslateTbl OBJECT-TYPE
    SYNTAX      SEQUENCE OF FwNatTranslateEntry
    ACCESS      not-accessible
    STATUS      mandatory
    DESCRIPTION ""
    ::= {fwNatAddrTransGrp 2}

```

```

aFwNatTranslateEntry OBJECT-TYPE
    SYNTAX      FwNatTranslateEntry
    ACCESS      not-accessible
    STATUS      mandatory
    DESCRIPTION ""
    INDEX      {fwTranslateSecIpAddr}
    ::= {fwNatTranslateTbl 1}

```

```

FwNatTranslateEntry ::= SEQUENCE
{
    fwTranslateSecIpAddr      DisplayString,
    fwTranslateSecIpAddrMask DisplayString
}

```

```

fwTranslateSecIpAddr OBJECT-TYPE
    SYNTAX      DisplayString (SIZE (0..100))
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "Defines the IP addresses to be excluded from NAT"
    ::= {aFwNatTranslateEntry 1}

```

```

fwTranslateSecIpAddrMask OBJECT-TYPE
    SYNTAX      DisplayString (SIZE (0..100))
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "The mask specifies the bits in the secured IP addr
    used to identify a range of IP addr."

```

```
::= {aFwNatTranslateEntry 2}
```

```
fwNatExcludeTbl OBJECT-TYPE  
SYNTAX SEQUENCE OF FwNatExcludeEntry  
ACCESS not-accessible  
STATUS mandatory  
DESCRIPTION ""  
::= {fwNatAddrTransGrp 3}
```

```
aFwNatExcludeEntry OBJECT-TYPE  
SYNTAX FwNatExcludeEntry  
ACCESS not-accessible  
STATUS mandatory  
DESCRIPTION ""  
INDEX {fwExcludeSecIpAddr}  
::= {fwNatExcludeTbl 1}
```

```
FwNatExcludeEntry ::= SEQUENCE  
{  
    fwExcludeSecIpAddr DisplayString,  
    fwExcludeSecIpAddrMask DisplayString  
}
```

```
fwExcludeSecIpAddr OBJECT-TYPE  
SYNTAX DisplayString (SIZE (0..100))  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION "Defines the IP addresses to be excluded from NAT"  
::= {aFwNatExcludeEntry 1}
```

```
fwExcludeSecIpAddrMask OBJECT-TYPE  
SYNTAX DisplayString (SIZE (0..100))  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION "The mask specifies the bits in the secured IP  
addr used to identify a range of IP addr."  
::= {aFwNatExcludeEntry 2}
```

```
fwNatMapTbl OBJECT-TYPE  
SYNTAX SEQUENCE OF FwNatMapEntry  
ACCESS not-accessible  
STATUS mandatory  
DESCRIPTION ""  
::= {fwNatAddrTransGrp 4}
```

```
aFwNatMapEntry OBJECT-TYPE  
SYNTAX FwNatMapEntry  
ACCESS not-accessible  
STATUS mandatory  
DESCRIPTION ""  
INDEX {fwMapSecIpAddr}  
::= {fwNatMapTbl 1}
```

```
FwNatMapEntry ::= SEQUENCE  
{  
    fwMapSecIpAddr DisplayString,  
    fwMapRegisteredIpAddr DisplayString  
}
```

```

fwMapSecIpAddr OBJECT-TYPE
    SYNTAX          DisplayString (SIZE (0..100))
    ACCESS          read-only
    STATUS          mandatory
    DESCRIPTION    "IP address to be translated into a specified
registered IP addr"
    ::= {aFwNatMapEntry 1}

fwMapRegisteredIpAddr OBJECT-TYPE
    SYNTAX          DisplayString (SIZE (0..100))
    ACCESS          read-only
    STATUS          mandatory
    DESCRIPTION    "IP address into which a specified secured IP address
should be translated."
    ::= {aFwNatMapEntry 2}

fwNatStatus OBJECT-TYPE
    SYNTAX          DisplayString (SIZE (0..20))
    ACCESS          read-only
    STATUS          mandatory
    DESCRIPTION    "The status of Network Address Translation"
    --            The possible values are active, deactive.
    ::= {fwNatAddrTransGrp 5}

fwNatLogStatus OBJECT-TYPE
    SYNTAX          DisplayString (SIZE (0..20))
    ACCESS          read-only
    STATUS          mandatory
    DESCRIPTION    "Logging status of Network Address Translation"
    --            The possible values are enabled, disabled
    ::= {fwNatAddrTransGrp 6}

```

END

Appendix D. ESP Specification for CDMF

If you have purchased the No Encryption version of the IBM eNetwork Firewall for AIX, this chapter does not apply.

This appendix describes the CDMF security transform for the IP Encapsulating Security Payload (ESP). See RFC1829 EPS DES_CBC for more details.

Keys

The secret CDMF key shared between the communicating parties is eight octets in length. This key is a 64-bit quantity used by the CDMF algorithm. The CDMF algorithm first runs a key-shortening algorithm to reduce the 64-bit value (56-bit key) to a 40-bit key. The 40-bit key is stored as 64 bits (eight octets). This 40-bit key is then used for encryption and decryption.

Initialization Vector

CDMF requires an Initialization Vector (IV) that is eight octets in length.

Each datagram contains its own IV. Including the IV in each datagram ensures that decryption of each received datagram can be performed, even when other datagrams are dropped, or datagrams are reordered in transit.

The method for selection of IV values is implementation dependent.

Data Size

The CDMF algorithm operates on blocks of eight octets. The CDMF algorithm is essentially the same as running DES_CBC with a weakened key (40 bits versus 56 bits). This often requires padding after the end of the unencrypted payload data.

Both input and output result in the same number of octets, which facilitates in-place encryption and decryption.

On receipt, if the length of the data to be decrypted is not an integral multiple of eight octets, then an error is indicated, as described in RFC-1825.

Payload Format

Figure 6 on page 182 shows the payload format.

Security Parameters Index (SPI)		
Initialization Vector (IV)		
Payload Data		
... Padding	Pad Length	Payload Type

Figure 6. Payload Format

Value Description

Security Parameters Index (SPI)

A 32-bit value identifying the Security Parameters for this datagram. The value must not be zero.

Initialization Vector (IV)

The size MUST be 64-bits. Octets are sent in network order with the most significant octet first (See RFC-1700.) It is the intent that the value not repeat during the lifetime of the encryption session key. Even when a full 64-bit IV is used, the session key should be changed at least as frequently as 2^{32} datagrams.

Payload Data

The size of this field is variable. Prior to encryption and after decryption, this field begins with the IP Protocol/Payload header specified in the Payload Type field. Note that in the case of IP-in-IP encapsulation (Payload Type 4), this will be another IP header.

Padding

The size of this field is variable. Prior to encryption, it is filled with unspecified implementation-dependent (preferably random) values, to align the Pad Length and Payload Type fields at an eight octet boundary. After decryption, it must be ignored.

Pad Length

This field indicates the size of the Padding field. It does not include the Pad Length and Payload Type fields. The value typically ranges from 0 to 7, but may be up to 255 to permit hiding of the actual data length. This field is opaque. That is, the value is set prior to encryption and is examined only after decryption.

Payload Type

This field indicates the contents of the Payload Data field, using the IP Protocol/Payload value. Up-to-date values of the IP Protocol/Payload are specified in the most recent "Assigned Numbers" (See RFC-1700.) This field is opaque. That is, the value is set prior to encryption, and is examined only after decryption. For example, when encrypting an entire IP datagram (Tunnel- Mode), this field will contain the value 4, which indicates IP-in-IP encapsulation.

Algorithm

In CDMF, the base CDMF encryption function is applied to the XOR of each plaintext block with the previous ciphertext block to yield the ciphertext for the current block. This provides for re-synchronization when datagrams are lost.

Encryption

Append zero or more octets of (preferably random) padding to the plaintext, to make its modulo 8 length equal to 6. For example, if the plaintext length is 41, 5 octets of padding are added.

Append a Pad Length octet containing the number of padding octets just added.

Append a Payload Type octet containing the IP Protocol/Payload value which identifies the protocol header that begins the payload.

Provide an Initialization Vector (IV) of the size indicated by the SPI.

Encrypt the payload with CDMF, producing a ciphertext of the same length.

Octets are mapped to CDMF blocks in network order with the most significant octet first. (See RFC-1700.) Octet 0 (modulo 8) of the payload corresponds to bits 1-8 of the 64-bit CDMF input block, while octet 7 (modulo 8) corresponds to bits 57-64 of the CDMF input block.

Construct an appropriate IP datagram for the target destination, with the indicated SPI, IV, and payload.

The Total/Payload Length in the encapsulating IP header reflects the length of the encrypted data, plus the SPI, IV, padding, Pad Length, and Payload Type octets.

Decryption

First, the SPI field is removed and examined. This is used as an index into the local Security Parameter table to find the negotiated parameters and decryption key.

The encrypted part of the payload is decrypted using CDMF.

The Payload Type is removed and examined. If it is unrecognized, the payload is discarded with an appropriate ICMP message.

The Pad Length is removed and examined. The specified number of pad octets are removed from the end of the decrypted payload, and the IP Total/Payload Length is adjusted accordingly.

The IP Header(s) and the remaining portion of the decrypted payload are passed to the protocol receive routine specified by the Payload Type field.

Security Considerations

Users need to understand that the quality of the security provided by this specification depends completely on the strength of the CDMF algorithm, the correctness of that algorithm's implementation, the security of the key management mechanism and its implementation, the strength of the key and upon the correctness of the implementations in all of the participating nodes.

Among other considerations, applications may wish to take care not to select weak keys, although the odds of picking one at random are low.

The cut and paste attack exploits the nature of all Cipher Block Chaining algorithms. When a block is damaged in transmission, on decryption both it and the following block will be garbled by the decryption process, but all subsequent blocks will be decrypted correctly. If an attacker has legitimate access to the same key, this feature can be used to insert or replay previously encrypted data of other users of the same engine, revealing the plaintext. The usual (ICMP, TCP, UDP) transport checksum can detect this attack, but on its own is not considered cryptographically strong. In this situation, user or connection oriented integrity checking is needed. (See RFC-1826.)

Appendix E. Obtaining Requests for Comments (RFCs)

Requests for comments (RFCs) are documents that present new protocols and establish standards for the Internet protocol suite. Hardcopies of all RFCs are available from the Network Information Center (NIC), either individually or on a subscription basis. You can obtain these documents from:

Government Systems, Inc.
Attn: Network Information Center
14200 Park Meadow Drive
Suite 200
Chantilly, VA 22021

You can access RFCs from this URL:

<http://www.cis.ohio-state.edu/hypertext/information/rfc.html>.

Online copies are available from the NIC using FTP to connect to `ds.internic.net`. You can transfer the files using the following format:

RFC:RFC*nnnn*.TXT
RFC:RFC*nnnn*.PS

Where:

nnnn Is the RFC number
TXT Is the text format
PS Is the PostScript format

The format for the RFC index is:

RFC:RFC-INDEX.TXT

Note: Many RFCs are only available in text format. Before requesting a PostScript file, first check the RFC Index to make sure the RFC is available in that format. You can also request online copies of the RFCs through the electronic mail, from the automated NIC mail server, by sending a message to `mailserv@ds.internic.net`. You must include one of the following commands in body of your note:

SEND RFC*nnnn*.TXT
or
SEND RFC*nnnn*.PS

Where:

nnnn Is the RFC number
TXT Is the text format
PS Is the PostScript format

For example, to request the text format of RFC 812, you would specify in the body of your note:

SEND RFC812.TXT

To request an online copy of the RFC index, include the following command in the body of your note:

SEND RFC-INDEX.TXT

Appendix F. Creating a Socks Configuration File

This chapter shows you what an AIX socks client file should look like.

The socks configuration file (`/etc/socks.conf`) for AIX systems is used by the socks client programs to permit or deny access through the firewall using the socks server, or to redirect a client request to a standard (non-socks) server.

Some socks client programs use this file to determine whether to use a direct or a socks server connection to a given destination host, and to exert access control based on the destination host, the requested service (port number on the destination host), and the effective user ID of the requesting local user.

Web browsers generally have their own socks configuration methods.

Every time a socks client has to make a network connection, the client checks the pending request against the file `/etc/socks.conf`, one line at a time. When the client finds a line with conditions that are matched by the request, the action specified on that line is taken. The remaining lines of file `/etc/socks.conf` are skipped. So the order of the lines in the file is extremely important; switch two lines and you might have entirely different results. If no matching line is found throughout the file, the request is denied.

Although there is an implied "deny all" at the end of the control file, you can supply an explicit "deny all" rule, for example:

```
deny 0.0.0.0 0.0.0.0.0 : /usr/bin/mail -s 'SOCKS: rejected %S from %u to %Z' root
```

Connection to address 127.0.0.1 (localhost) and 0.0.0.0 (broadcast) is always done directly, so there is no need to specify those in `/etc/socks.conf`.

Notes:

1. Each line in the file can be up to 1023 characters long.
2. Spaces and tabs separate the fields.
3. Comment lines start with the character #.

The parameter options for the socks file are:

deny

Tells the socks clients when to reject a request.

direct

Tells the socks clients when to use a direct connection.

sockd

Tells the socks clients when to use a socks server connection and, optionally, which socks server it should try.

@=serverlist

Can be used only in a sockd rule. It names one or more socks servers that the socks client should try to use (in the indicated order). Only commas can be used as separators, no spaces or tabs are allowed in the list. Domain names of the servers can be used in the list, though it is probably more prudent to specify IP addresses. If this field is omitted, the client program will use the default socks server, which is determined by the environment variable `SOCKS_SERVER` if it exists. If that is not set, the client will try to use the name compiled into the socks client itself.

***=userlist**

Consists of one or more user IDs or file names, separated by commas. (No spaces or tabs are allowed in the list.)

- The user IDs should be IDs of users on the local host, not those on the destination host or the socks server host.
- The file names must be full path names (with the leading /). The specified files contains the user IDs, listed one or more to a line. You can separate the IDs with any combination of blanks, tabs, and commas.
- If the *= userlist field is omitted, the line applies to all user IDs.

destination_address destination_mask

These operands specify the destination IP address or the range of destination IP addresses. Specify both in the usual dotted form, for example 129.1.2.3. Bits in `destination_mask` that are set to 0 indicate the bit positions which should be masked off (ignored) during comparison of `destination_address` and the actual destination IP address.

For example, if you specify 255.255.255.255 in the `destination_mask` field, the actual destination address must match exactly the address specified in the `destination_address` field.

But if you specify 0.0.0.0 in the `destination_mask` field, any destination address will match.

Note: This is the way subnet masks are interpreted in TCP/IP, but is the opposite of how the address masks are used in a router's access-lists.

log_op destination_portnumber

The `log_op` is an operation field, and can be one of these:

- eq** equal to
- neq** not equal to
- lt** less than
- gt** greater than
- le** less than or equal to
- ge** greater than or equal to

The `destination_portnumber` is a destination port.

The socks server uses the logical operation and the port number to compare to the port number in a request. The destination port in the request, and the `destination_portnumber` field must relate as stated by the `log_op`.

For example, if `log_op` is EQ and `destination_port` is 23, then the incoming request is allowed ONLY if it is for port 23. If you omit this pair, the rule applies to all ports.

command

Enter a colon (:) followed by a command to be executed when the conditions on that line are satisfied. The following substitutions occur before the string is presented to the Bourne shell for execution:

- %A** Replaced by the client host's domain name if known, by its IP address otherwise
- %a** Replaced by the client host's IP address

%c	Replaced by "connect" or "bind"
%p	Replaced by the process id of the client program
%S	Replaced by the service name (ftp) if known, by the destination port number otherwise
%s	Replaced by the destination port number
%U	Replaced by the user ID at login
%u	Replaced by the effective user ID
%Z	Replaced by the destination host's domain name if known, by its IP address otherwise
%z	Replaced by the destination host's IP address
%%	Replaced by a single %

Several shell commands can be strung together "|" or ";".

Example of a sockd Rule

Consider this sample rule:

```
sockd @=1.2.3.4 *=boss,root 11.12.13.14 255.255.255.255 eq 23
```

To match the condition indicated in this line, a request must come from a local user whose effective ID is either boss or root. The destination IP address must be 11.12.13.14 exactly. The destination port must be 23. In that case, connection to host 11.12.13.14 should be done via a socks server on host 1.2.3.4.

Appendix G. The Crontab Command

The crontab command submits, edits, lists, or removes cron jobs. A cron job is a command run by the cron daemon at regularly scheduled intervals.

crontab syntax

```
crontab [-e | -l | -r | -v | File ]
```

- e** Edits a copy of your crontab file or starts an editing session if you do not already have a crontab file. Each entry must be in a form acceptable to the cron daemon. When editing is complete, the entry is installed as your crontab file. The editing session is started using the editor specified by the EDITOR environment variable.
The default editor is vi.
- l** Lists the contents of your crontab file.
- r** Removes an existing crontab file from the crontab directory.
- v** Lists the status of your cron jobs.
- File** Allows you to create your own crontab files.

When you finish creating entries and exit the file, the crontab command copies the file into the **/var/spool/cron/crontabs** directory and names it with your current username. If a file with your name already exists in the crontabs directory, the crontab command overwrites the existing name.

Alternatively, you can create a crontab file by specifying the File parameter. If the file already exists, it must be in the format the cron daemon expects. If the file does not exist, the crontab command invokes the editor. If the EDITOR environment variable exists, the command invokes the editor it specifies. Otherwise, the crontab command uses the vi editor.

The cron daemon runs commands according to the crontab file entries. Unless you redirect the output of a cron job to standard output or error, the cron daemon mails you any command output or error. If you specify a cron job incorrectly in your crontab file, the cron daemon does not run the job.

The cron daemon examines crontab files only when the cron daemon is initialized. When you make changes to your crontab file using the crontab command, a message indicating the change is sent to the cron daemon. This eliminates the overhead of checking for new or changed files at regularly scheduled intervals.

The **/var/adm/cron/cron.allow** and **/var/adm/cron/cron.deny** files control which users can use the crontab command. A root user can create, edit, or delete these files. Entries in these files are user login names with one name to a line. If your login ID is associated with more than one login name, the crontab command uses the first login name that is in the **/etc/passwd** file, regardless of which login name you might actually be using.

Here is a quick method for setting up a crontab. To learn more about the AIX crontab function, issue "**man crontab**" from the AIX command line.

To set up a crontab that will compress and archive all log files (that have been configured to be archived) every Sunday at 2am, follow these steps:

1. Start an editor session on the crontab file by issuing the **crontab -e** command.

Note: This should bring up an editor session using the editor defined by your \$EDITOR variable. If you wish to use another editor, you can either change the value of the \$EDITOR variable or issue "crontab -1>tempcron". You can then edit the tempcron file and issue "crontab tempcron" to activate your changes to the file.

2. Each crontab file entry contains six fields separated by spaces or tabs in the following form:

```
minute hour day_of_month month weekday
```

These fields accept the following values:

minute:

0 through 59

hour: 0 through 23

day_of_month:

1 through 31

month:

1 through 12

weekday:

0 through 6 for Sunday through Saturday

To run the **fwlogmgmt** command every Sunday at 2 am, add the following line to the bottom of the crontab file:

```
0 2 * * 0 /usr/bin/fwlogmgmt -1
```

Your crontab file should look something like:

```
-----  
#(c) COPYRIGHT International Business Machines Corp. 1989,1994  
#All Rights Reserved  
#Licensed materials - Property of IBM  
#  
#US Government Users Restricted Rights - Use, duplication or  
#disclosure restricted by GSA ADP Schedule Contract with IBM Corp.  
#  
#0 3 * * * /usr/sbin/skulker  
#45 2 * * 0 /usr/lib/spell/compress  
#45 23 * * * ulimit 5000; /usr/lib/smdemon.cleau > /dev/null  
0 11 * * * /usr/bin/errclear -d S,0 30  
0 12 * * * /usr/bin/errclear -d H 90  
0 2 * * 0 /usr/bin/fwlogmgmt -1  
-----,
```

3. Save the file to activate the changes.

Bibliography

For additional information about security on the Internet, visit the IBM Firewall home page at <http://www.software.ibm.com/enetwork/firewall>.

Information in IBM Publications

Other IBM sources of information on firewalls, Internet security, and general security topics are listed here.

Firewall Topics

The following documents are available on the IBM Firewall CD-ROM and the IBM eNetwork Firewall home page.

- *IBM eNetwork Firewall User's Guide*, GC31-8419
- *IBM eNetwork Firewall Reference*, SC31-8418
- *Protect and Survive Using the IBM Firewall 3.1 for AIX*, SG24-2577

Internet and World Wide Web Topics

- *A Guide to the Internet Connection Servers*, SG24-4805
- *Accessing CICS Business Applications from the World Wide Web*, SG24-4547
- *Accessing OS/390 OpenEdition MVS from the Internet*, SG24-4721
- *Accessing the Internet*, SG24-2597
- *Building the Infrastructure for the Internet*, SG24-4824
- *Cool Title about the AS/400 and Internet*, SG24-4815
- *The Domino Defense: Security in Lotus Notes and the Internet*, SG24-4848
- *Examples of Using MQSeries on WWW*, SG24-4882
- *How to Secure the Internet Connection Server for MVS/ESA*, SG24-4803
- *Lotus Domino Server Release 4.5 on AIX Systems: Installation, Customization, and Administration*, SG24-4694
- *Netscape Proxy Server*, SK2T-7444
- *Running CICS Transactions through the Web: The CICS Internet Gateway to VSE/ESA*, SG24-4799

- *Safe Surfing: How to Build a Secure World Wide Web Connection*, SG24-4564
- *Teach Yourself CGI Programming with PERL in a Week*, SR23-7343
- *Using the Information Super Highway*, GG24-2499
- *World Wide Web Access to DB2*, SG24-4716

General Security Topics

- *The Basics of IP Network Design*, SG24-2580
- *Elements of Security: AIX V4.1*, GG24-4433
- *Enterprise-Wide Security Architecture and Solutions Presentation Guide*, SG24-4579
- *HACMP/6000 Customization Examples*, SG24-4498
- *IBM Global Network (IGN) Security Policy*, GC34-2206
- *IBM Security Architecture: Securing the Open Client/Server Distributed Enterprise*, SC24-8135
- *IBM Systems Monitor: Anatomy of a Smart Agent*, SG24-4398
- *Security Overview of Open Systems Networking*, GG24-3815
- *Systems Monitor for AIX User's Guide*, SC31-8173
- *TCP/IP Tutorial and Technical Overview*, GG24-3376

Information in Industry Publications

These industry publications pertain to sendmail, TCP/IP and UNIX:

- Albitz, Paul, and Cricket Liu. *DNS and BIND*. Sebastopol, CA: O'Reilly and Associates, 1997. (ISBN: 1-56592-236-0)
- Costales, Brian with Eric Allman. *Sendmail*. O'Reilly and Associates, Inc. (ISBN: 1-56592-222-0)
- Hunt, Craig. *TCP/IP Network Administration*. O'Reilly and Associates, Inc. (ISBN: 0-937175-82-X)
- Nemeth, Snyder, et al. *UNIX System Administration Handbook*. Prentice Hall. (ISBN: 0-13-151051-7)

These industry publications pertain to firewalls and security on the Internet:

- Ahuja, Vijay. *Network and Internet Security*. Boston: Academic Press Professional, 1996. (ISBN: 0120455951)
- Ahuja, Vijay. *Secure Commerce on the Internet*. Boston: Academic Press Professional, 1997. (ISBN: 0120455978)
- Anderson, Bart, et al. *The Waite Group's UNIX Communications and the Internet*. Indianapolis, IN: Sams Pub., 1995. (ISBN: 0672305372)
- Atkins, Derek, et al. *Internet Security: Professional Reference*. Indianapolis, IN: New Riders Publishing, 1996. (ISBN: 1562055577)
- Chapman, D. Brent, and Elizabeth D. Zwicky. *Building Internet Firewalls*. Sebastopol, CA: O'Reilly and Associates, 1995. (ISBN: 1565921240)
- Cheswick, Willam R., and Steven M. Bellovin. *Firewalls and Internet Security*. New York: Addison-Wesley, 1994. (ISBN: 0201633574)
- Cooper, Frederic J., et al. *Implementing Internet Security*. Indianapolis, IN: New Riders Publishing, 1995. (ISBN: 1562054716)
- Curry, David. *UNIX System Security: Guide for Users and Systems Administrators*. Sebastopol, CA: O'Reilly and Associates, 1994. (ISBN: 0201563274)
- Garfinkel, Simson, and Gene Spafford. *Practical UNIX Security*. Sebastopol, CA: O'Reilly and Associates, 1991. (ISBN: 0937175722)
- Garfinkel, Simson, and Gene Spafford. *Practical UNIX and Internet Security*. Sebastopol, CA: O'Reilly and Associates, 1996. (ISBN: 1565921488)
- Hare, Chris, and Karanjit Siyan. *Internet Firewalls and Network Security*. Indianapolis, IN: New Riders Publishing, 1996. (ISBN: 1562056328)
- Randall, Neil. *Teach Yourself the Internet in a Week*. Indianapolis, IN: Sams.Net, 1995. (ISBN: 0672307359)
- Stallings, William. *Internet Security Handbook*. Foster City, CA: IDG Books, 1995. (ISBN: 0077092546)
- Stevens, W. Richard. *TCP/IP Illustrated*. Reading, MA: Addison-Wesley, 1994. (ISBN: 0201634953)

Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make them available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Subject to IBM's valid intellectual property or other legally protectable rights, any functionally equivalent product, program, or service may be used instead of the IBM product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the responsibility of the user.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
500 Columbus Avenue
Thornwood, NY 10594
USA

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
P.O. Box 12195
3039 Cornwallis Road
Research Triangle Park, NC 27709-2195
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement.

This document is not intended for production use and is furnished as is without any warranty of any kind, and all warranties are hereby disclaimed including the warranties of merchantability and fitness for a particular purpose.

This product includes software developed by the University of California, Berkeley and its contributors.

Trademarks

The following terms are trademarks of the IBM corporation in the United States or other countries or both:

- AIX
- AIXwindows
- AIX/6000
- Common User Access

- DB2
- eNetwork
- HACMP
- IBM
- OS/2
- RS/6000
- RISC System/6000

Microsoft, Windows, Windows NT, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Java and HotJava are trademarks of Sun Microsystems, Inc.

Other company, product, and service names, which may be denoted by a double asterisk (**), may be trademarks or service marks of others.

Glossary

You can access the IBM Software glossary at:
<http://www.networking.ibm.com/nsg/nsgmain.htm>.

Index

Special Characters

, firewall daemons 1
(MIB), SNMP Management Information Base 155
(MKKF), Using the Make Key File Utility 79
(RFCs), Requests for comments 185

A

a_alert.tbl 46
a key file, Creating 79
Adapters 1
Address, MAP Secured IP 17
Addresses, Exclude Secured IP 17
Addresses, Translate Secured IP 17
ADMIN_ALERT 52
Administration Functional Groups 39
alert log 13
and FWUSERAU Specifications, FWUSERPT 65
Authentication, User-Supplied 65
authentication methods 65

B

bibliography 193

C

command, crontab 191
Command Line Interface 1
configuration file, socks 187
Configuration Server 2
Creating a Key File 79
crontab command 191

D

daemons, firewall 1
DB2 47
DB2/6000 or DB2/2 43
DNS Problems 89
Domain Name Services 5

E

Enterprise Firewall Manager 6
ESP Specification for CDMF 181
Exclude Secured IP Addresses 17

F

f_info.tbl 46
f_match.tbl 46
f_rule.tbl 46
f_stat.tbl 46
file, socks configuration 187
File Management, Log 12
FILTER_ACTIVE_RULE 52

FILTER_INFO 52
FILTER_MATCH 52
FILTER_STATUS 52
Filters 8
firewall daemons 1
firewall log 13, 43
Functional Groups, Administration 39
Fundamental Parameters 36
fw_ftp_authenticate 68, 70
fw_prompt 68
fw_ret_struct 68, 69
fw_tn_authenticate 68
fwadapter 1
fwar2asc 43, 45
fwconns 2
fwfilter 8
fwfschk 8
fwice 11, 98
fwimport.dat 43
fwlog 12
fwlogmon 14
fwlogtbl 43, 44
fwlogtxt 43, 44
fwmail 16
fwmanager 6
fwnat 17
fwqrysmp.dml 43
fwschema.ddl 43, 47
fwtransfer 7
fwuser 35
fwuser.h 68
fwuser.o 73
fwuserau 66
FWUSERAU Specifications, FWUSERPT and 65
fwuserpt 66
FWUSERPT and FWUSERAU Specifications 65

G

Generating Messages 44
Groups, Administration Functional 39

H

hardening 153
HTTP proxy 11

I

IBM Firewall, Network Management with the 75
Idle Proxy Parameters 37
IP Address, MAP Secured 17
IP Addresses, Exclude Secured 17
IP Addresses, Translate Secured 17

K

key file, Creating a 79

L

licensing agreement 195
log, firewall 43
Log Facilities 97
Log File Management 12
Log Management 98
Log Monitor 14
Login Shells 36

M

makefile 72
Management, Log File 12
Many-to-One Registered Address 17
MAP Secured IP Address 17
messages 101
Messages, Generating 44
methods, authentication 65

N

Name Services, Domain 5
NAT 93
NAT_INFO 52
nat_info.tbl 46
Network Address Translation 17
Network Management with the IBM Firewall 75
Network Object Groups 21
Network Objects 19

P

p_ftp.tbl 46
p_http.tbl 46
p_info.tbl 46
p_login.tbl 46
p_stat.tbl 46
PAGER_INFO 52
Parameters, Fundamental 36
Parameters, Idle Proxy 37
proxy, HTTP 11
PROXY_FTP 52
PROXY_HTTP 52
PROXY_INFO 52
PROXY_LOGIN 52
Proxy Parameters, Idle 37
Proxy Servers 93
PROXY_STATUS 52

Q

Queries, Sample 48

R

references 193
Registered Address, Many-to-One 17
report utilities 43
Report Utilities 97
Report Utilities Usage 43
Requests for comments (RFCs) 185

Ret_code 69

S

s_ftp.tbl 46
s_info.tbl 46
Sample Queries 48
Secure Key 66
Secured IP Address, MAP 17
Secured IP Addresses, Exclude 17
Secured IP Addresses, Translate 17
Security Policies 28
SERVER_INFO 52
server_info.tbl 46
Services 28
Services, Domain Name 5
SESSION 52
session.tbl 46
Shells, Login 36
Simple Network Management Protocol (SNMP) 75
SMIT 48
SNMP Management Information Base (MIB) 155
SNMP trappable events 76
socks configuration file 187
SOCKS_FTP 52
SOCKS_INFO 52
Socks Rules 31
Specifications, FWUSERPT and FWUSERAU 65
SQL Tables 52
SSL_INFO 52
ssl_info.tbl 46
SU 52
su.tbl 46
support for ftp, user-supplied 67
support for telnet, user-supplied 67

T

t_cntxt.tbl 46
t_policy.tbl 46
t_stat.tbl 46
Tables, SQL 52
Traffic Control 92
Translate Secured IP Addresses 17
trappable events, SNMP 76
Troubleshooting and Testing 87
TUNNEL_CONTEXT 52
TUNNEL_POLICY 52
TUNNEL_STATUS 52
Tunnels 32

U

URLs 193
User-Supplied Authentication 65
User-Supplied Iteration Support 66
user-supplied support for ftp 67
user-supplied support for telnet 67
Using the Make Key File Utility (MKKF) 79
utilities, report 43

W

Web page 193
Win 95 Remote IPSec Client 96

Readers' Comments — We'd Like to Hear from You

**IBM eNetwork Firewall for AIX
Reference
Version 3 Release 2.2**

Publication No. SC31-8418-02

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Complete	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to find	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to understand	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Well organized	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Applicable to your tasks	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? Yes No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Name

Address

Company or Organization

Phone No.



Cut or Fold
Along Line

Fold and Tape

Please do not staple

Fold and Tape



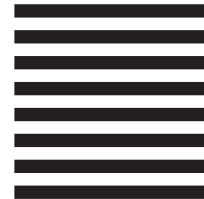
NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation
Information Development
Department CGMD / Bldg 500
P.O. Box 12195
Research Triangle Park, NC
27709-9990



Fold and Tape

Please do not staple

Fold and Tape

Cut or Fold
Along Line



Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

SC31-8418-02

