IBM eNetwork™ Firewall for Windows NT®

Reference

Version 3 Release 3

SC31-8659-02

IBM eNetwork™ Firewall for Windows NT®

Reference

Version 3 Release 3

SC31-8659-02

Note

Before using this information and the product it supports, be sure to read the general information under "Notices" on page 169.

Third Edition (March 1999)

This edition applies to Version 3 Release 3 of the IBM eNetwork Firewall for Windows NT (product number 5765-C16). This edition replaces SC31-8659-01.

Contents

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 on a Windows NT® machine. To use client programs such as Telnet or FTP, please see the user's guide for your TCP/IP client programs.

Technical changes to the text are indicated by a vertical line to the left of the text.

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.

A recommended book on TCP/IP that covers netstat, arp, ifconfig, ping, nslookup, DNS, sendmail, routing, and much more is *TCP/IP Network Administration*. See "Bibliography" on page 167 for more details.

A recommended book for those performing administration is the *Windows NT Server 4.0 Administrator's Bible*. See "Bibliography" on page 167 for more details.

Enhancements

The IBM eNetwork Firewall V3R3 for Windows NT offers several enhancements:

1. Virtual Private Networks with 3DES/DES/CDMF encryption and HMAC-MD5/HMAC-SHA authentication and compliant with the latest Internet Engineering Task Force RFCs
2. Secure Mail Proxy
3. Setup wizard
4. National Language Support for German

Virtual Private Networks

The IBM eNetwork Firewall V3R3 provides support for manually configured VPN tunnels. A Virtual Private Network (VPN) is an extension of an enterprise's private intranet across a backbone network, which typically will be a public backbone such as the Internet. A VPN allows you to create a secure connection to protect your data while it is in transit over the backbone. The VPN tunnel uses the open IPSec security standards to protect your data from modification or disclosure while it is travelling between firewalls. Your data will flow within a VPN tunnel, which can provide data origin authentication, confidentiality, and integrity checking on every packet. IPSec protocols can keep your data private, hiding it from any eavesdroppers on the public network. Packet filtering in the firewall can be used in conjunction with IPSec technologies to further protect your intranets from unwanted intrutions.

VPNs can be created by manually configuring VPN tunnels between pairs of IBM Firewalls or between an IBM Firewall and any other device (client, router, server, or firewall) that supports the latest open IPSec standards. Encryption support can include DES, 3DES, and CDMF. Authentication support includes HMAC-MD5 and HMAC-SHA. Filters for your VPN tunnels can be customized, or you can choose a default set of filter rules.

VPNs have a dynamic filters option, which saves the firewall administrator time because he or she does not have to create filter rules. If you select the dynamic filter rules option, filter rules are generated each time a tunnel is activated.

Secure Mail Proxy

The IBM Firewall Secure Mail Proxy is a new enhanced component that replaces SafeMail. The Secure Mail Proxy provides a gateway for SMTP traffic. It relays messages from the secure mailserver to the nonsecure side, hiding sensitive domain names as it goes. It relays messages from the nonsecure side in to the secure mail domain and insulates the secure network from attacks.

The Secure Mail Proxy acts as a real-time gateway between two or more e-mail domains. In contrast with a traditional SMTP relay, messages are not stored on the Firewall before being forwarded to the destinations. The SMTP conversation is interpreted as it happens, and the SMTP proxy conversation is proxied on to each of the necessary destination servers, command by command.

KeyWorks

The IBM Firewall uses a secure cryptographic toolkit called KeyWorks to provide the key recovery function. KeyWorks performs self-checking so that it can guarantee the authenticity of each of its components, through the use of digital certificates. KeyWorks' components expire when the certificates expire. After the installation of the Firewall is completed, a message is logged that displays the expiration date of the KeyWorks' components. Prior to expiration, it is necessary to refresh the authentication certificates.

Key recovery may be required to comply with US export regulations or your country's import regulations. It is intended for law enforcement purposes only. Key Recovery is a technique that permits recovery of encrypted data as it flows through VPN tunnels. Before strong encryption can be used, an encrypted block of data (called the key recovery block) containing the key to be used, is transmitted to the firewall at the tunnel partner. The key recovery block (KRB) must be acknowledged by the receiver. After the acknowledgement successfully reaches the sender, the key can be used.

Setup Wizard

A wizard has been provided to aid the user with the initial configuration of the IBM eNetwork Firewall. This setup wizard enables a user, who does not have extensive knowledge of the Firewall, to have a basic Firewall configuration up and running quickly after installation.

National Language Support for German

National language support for German is offered in addition to Brazilian Portugese, English, French, Italian, Japanese, Korean, simplified Chinese, Spanish, and traditional Chinese.

Terminology

You can access the IBM Software glossary at: **http://www.networking.ibm.com/nsg/nsgmain.htm**.

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. Or you can access the following web site: http://www.software.ibm.com/enetwork/support/ and specify the country where service is required.

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

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 services dynamically update their behavior when their configuration files change. Some require an update subcommand. An `update` subcommand is provided for those firewall services that require an instruction.
- Only primary firewall administrators can execute programs from the command line.
- Because of the complexity and file interdependencies, **do not directly edit any configuration files**.

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 like this:

```
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 65.

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.

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

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=]
        [threadpoolsize=]
        [logging={on|off}]
        [authenticate={all|new|none}]
        [authenticatetimeout=]
        [maxpersistrequests=]
        [persisttimeout=]
```

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.

threadpoolsize
Sets the fixed number of threads active at one time. The proxy holds new requests until another request finishes and threads become available. In general, the more power a machine has, the bigger the value you should use for this parameter.

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

authenticate
The level of users to authenticate. Values are all, none, or new.

authenticatetimeout
The time to wait for a client request after establishing a persistent connection.

maxpersistrequests
The maximum number of requests to receive on a persistent connection.

persisttimeout
Time to keep a persistent connection.

Interfaces

Secure interfaces connect the IBM Firewall host to the network of hosts in your internal network, the network that you want to protect. **You must have at least one secure interface for your firewall to work.** Nonsecure interfaces connect the IBM Firewall to one or more outside networks or to the Internet. The IBM Firewall must have at least one nonsecure interface.

This command lists the firewall's network interfaces. An administrator must have the authority to administer interface functions to issue this command.

```
fwinterface cmd=list
            [addr=x.x.x.x]
```

See the Administration chapter of the *IBM eNetwork Firewall User's Guide* for more information on administrator authority.

The parameter definitions are:

addr=x.x.x.x
Lists all of the network interfaces that have been configured to the firewall and identifies each as being either a secure or a nonsecure interface. A name could also be identified. If the optional addr parameter is specified, only that interface is listed. If a dotted-decimal IP address is provided for addr the list will contain the address, state, and name of only the specified address, assuming it has been configured to the firewall.

This command allows you to define your network interfaces to the firewall. An administrator must have the authority to administer interface functions to issue this command.

```
fwinterface cmd=change
            addr=x.x.x.x
            {state={secure|nonsecure} and/or
            name="interface name"}
```

The parameter definitions are:

addr=x.x.x.x
Contains the dotted-decimal address of the interface to be changed. If that interface is not defined to the firewall, an error will be reported.

state={secure | nonsecure}
Contains one of two keywords ″secure″ or ″nonsecure″ which categorize the network that is attached to the specified interface.

name Is a meaningful name identifying the interface or the network to which it is attached. Spaces can be included, providing that they are properly double-quoted.

Although both the state and name parameters are optional, one of them must be specified.

Log Archiver

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

```
fwlogmgmt -1 or fwlogmgmt -a
```

It is useful to put this command in a Windows NT Scheduled Service. 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={firewall|alert|socks|audit}
       priority={debug|info|warning|err|crit}
       logfile=_LogFilePathName_
 [logtime=_DaysToKeepInLogFile_
       arcfile=_ArchivePathName_
       arctime=_DaysToKeepInArchive_]
```

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
- adminaudit (local0) - administrative audit log

Valid values for **priority**:

- debug
- info
- warning

- err
- crit

The logfile parameter indicates where the firewall logging entries should be sent. The valid value for logfile is a fully qualified file name (with the format (drive:\directory) indicating the file to which the log entries should be written.

Note: Files identified for the `alert log` or `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.

The arcfile, logtime, and arctime parameters are optional, and are only valid when the logfile parameter specifies a file name. All three 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 6.

By default the firewall uses these parameters to indicate where to store archive log records and how often the archiving should take place. You need to specify these three parameters to enable archiving.

The archiving function can be replaced by writing a firewall archiving plug-in. See “Chapter 3. Log Archiver Plug-in Software Development Kit” on page 53.

The **arcfile** parameter must contain a fully qualified path.

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

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

To change log facilities, issue the following command.

```
fwlog cmd=change
       index=_index_
       facility={firewall|alert|socks|audit}
       priority={debug|info|warning|err|crit}
       logfile=_LogFilePathName_
 [logtime=_DaysToKeepInLogFile_
       arcfile=_ArchivePathName_
       arctime=_DaysToKeepInArchive_]
       force={yes|no}
```

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
      force={yes|no}
```

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=CommandToExecute
         [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=NumberofFailures
         time=Time
         pager=Y|N
         [comment=]
```

```
fwlogmon cmd=change
         type=single|multi|host
         [count=NumberofFailures]
         [time=time]
         [pager=Y|N]
         [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=msgtag
         count=NumberofFailures
         time=time
         pager=Y|N
         [comment=]
```

```
fwlogmon cmd=change
         type=msg
         tag=msgtag
         [count=NumberofFailures]
         [time=time]
         [pager=Y|N]
         [comment=]
```

```
fwlogmon cmd=delete
         type=msg
         tag=msgtag
```

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.

Use the fwsecuremail command to modify the overflow server configuration. The refresh command must be used to trigger the server to re-read the configuration after changes are made.

```
fwsecuremail cmd=list
```

```
fwsecuremail cmd=list
```

```
fwsecuremail cmd=change
        [overflow_host]
        [overflow_port]
```

```
fwsecuremail cmd=refresh
```

```
fwsecuremail cmd=shutdown
```

The parameter definitions are:

overflow_host
Can either be a host name or a dotted-decimal address enclosed in square brackets.

overflow_port
An integer. Overflow_host and overflow_port together identify the overflow server.

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 delete a many-to-one entry in the NAT configuration file use the following syntax:

```
fwnat cmd=delete
        index=
```

The parameter definition is:

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.

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

```
fwnat cmd=delete
        index=
```

The parameter definition is:

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.

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

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

```
fwnat cmd=delete
        index=
```

The parameter definition is:

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.

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=
        pagerid=
        message=
```

To modify pager parameters, issue the following command.

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

The parameter definitions are:

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

modem
A name for the modem, as defined in the modems database (through the `fwmodem` command).

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=
          pagmethod={TAP}
          [password=]
          length=
          baud={300|600|1200|2400|4800|9600|19200|38400}
          parity={none|even|odd}
          databits={7|8}
          stopbits={1|2}
```

To modify carrier parameters, issue the following command.

```
fwcarrier cmd=change
          carrier=
          [dial=]
          [pagmethod=]
          [password]
          [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.

pagmethod
The value must be TAP.

password
This is optional unless needed for the carrier service.

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
        modem=
```

To add a modem, issue the following command.

```
fwmodem cmd=add
        modem=
        comport=
        initsting=
        outsideline=
```

To modify modem parameters, issue the following command.

```
fwmodem cmd=change
        modem=
        [comport=]
        [initstring=]
        [outsideline=]
```

To delete a modem, issue the following command.

```
fwmodem cmd=delete
        modem=
```

The parameter definitions are:

modem
A name for the modem.

comport
The serial COM port to which the modem is attached. The modem on this COM port must not be defined to your Windows NT system.

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.

Testing Pager Configuration

To ensure that you have correctly configured your active pager, use the following command.

```
pager
    carrier=
    modem=
    ID=
    msg=
```

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 ROOTDIR\config\pager.cfg 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 pager.cfg files you need
- Copy the configuration file that you want to activate back to ROOTDIR\config\pager.cfg

If you are trying to handle shift changes, set up a scheduled job using the Windows NT at command to automatically copy the appropriate configuration file at the start of each shift.

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}]
       [secftp=SecureFTPauthentication]
       [remftp=NonSecureFTPauthentication]
       [secauth=SecureTelnetAuthentication]
       [remauth=NonSecureTelnetAuthentication]
       [secadmin=SecureAdminAuthentication]
       [remadmin=NonSecureAdminAuthentication]
       [secsocks=SecureSocks]
       [remsocks=NonSecureSocks]
       [sechttp=SecureHTTP]
       [histexpire=HistoryExpiration]
       [histsize=HistorySize]
       [loginretries=LoginRetries]
       [maxage=MaxAge]
       [maxexpired=MaxExpiredAge]
       [maxrepeats=MaxRepeatChars]
       [minalpha=MinAlphaChars]
       [mindiff=MinDifferentChars]
       [minlen=MinLength]
       [minother=MinNonAlphaChars]
       [pwdwarntime=PasswordWarnTime]
       [pwuserchng={yes|no}]
       [pwlocked={yes|no}]
       [fg_addtrans={yes|no}]
       [fg_all={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_user={yes|no}]
       [fg_traffic={yes|no}]
       [fg_vpn={yes|no}]
```

Fundamental Parameters

username

Login name for this user.

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.

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
- NT–NT logon password
- sdi–SDI
- user–user-supplied authentications
- userauth2–user-supplied authentications
- userauth3–user-supplied authentications

secftp Authentication method for FTP logins from a secure interface. Valid values are deny, permit, password, NT, sdi, user, userauth2, and userauth3. The default is deny.

remftp
Authentication method for FTP logins from a nonsecure interace. Valid values are deny, permit, password, NT, sdi, user, userauth2, and userauth3. The default is deny.

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

remauth
Authentication method for telnet logins from a nonsecure interface. Valid values are deny, permit, password, NT, sdi, user, userauth2, userauth3. The default is deny.

secadmin
Authentication method for Firewall Configuration Client logins from a secure interface. Valid values are deny, permit, password, NT, sdi, user, userauth2, and userauth3. The default is deny for proxy users and NT for Primary Firewall administrators.

remadmin
Authentication method for Firewall Configuration Client logins from a nonsecure interface. Valid values are deny, permit, password, NT, sdi, user, userauth2, userauth3. The default is deny for proxy users and NT for Primary Firewall users.

secsocks
Socks5 authentication method for Socks client connections coming from the secure side of the firewall. Valid values are deny, permit, password, NT, sdi, user, userauth2, and userauth3.

The default is deny.

remsocks
Socks5 authentication method for Socks client connections coming from the nonsecure side of the firewall. Valid values are deny, permit, password, NT, sdi, user, userauth2, and userauth3.

The default is deny.

sechttp
Authentication method for HTTP requests from a secure interface. Valid values are deny, permit, password, NT, sdi, user, userauth2, and userauth3.

SDI is supported but the user will be prompted for a password instead of an SDI passcode. The user should enter his or her SDI passcode.

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

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.

userchng
Determines how the administrator change flag will be set in the user database. A value of yes sets the administrator change flag which requires the user to change his password the first time he logs on. No

is the default. This parameter is only valid if the password=yes and pwdvalue=″ parameters are supplied.

pwlocked
Indicates whether the password has been locked. This is set to yes when the maximum number of failed logins is exceeded or when the password has not been used for the number of weeks specified in maximum time before lockout.

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 sets `pwlocked` to no.

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

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

fg_addrtrans
Enter yes if this administrator is allowed to administer network address translation. 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_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® . 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:

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

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

fwlogcvrt
You might need to use this program if you use other vendors' reporting tools that support the IBM Firewall for AIX® logs. This program converts a Windows NT firewall log format to an AIX firewall log format. This enables other vendors' reporting tools to operate as before except that new messages might not be recognized.

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 and other database and file systems. Their simple format should allow conversion to other formats, if necessary.

Report Utilities Usage

This information explains how to use report utilities from the command line. 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" on page 29 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.
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:Amsg_num; msg_ID;var_1;...;var_n;
```

where

- The first three fields, **date, time, and firewall-name** are added by the firewall logging facility.
- **year** is the four-character year.
- **pid** is the thread ID to which the entry applies.
- **Amsg_num** is a sequential integer which the Report Utilities use to access the appropriate, translated message text from the fw_log.cat file. The numeric msg_num is immediately preceded by a log level indicator letter (A). This indicator distinguishes both the platform that orginated the log and any differences in log format.
- **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 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.

Use the command `fwlogcvrt` to convert from this Windows NT release's log format to that of an AIX log. You might need to do this to use other vendor reporting tools that support the IBM Firewall for AIX logs. The conversion will remove the 'A' log level indicator that precedes the msg_num and inserting two blank characters around the colon between the firewall_name and the year.

The parameters include:

input Standard input redirected from a Windows NT Firewall log.

output
Standard output, which can be redirected to a file.

fwlogcvrt syntax

```
fwlogcvrt
```

Example:

```
fwlogcvrt < fw980212.log >logcvrt.out
```

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 < my.log | find "ICA0"
```

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

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.

output
File names:

a_alert.tbl

f_rule.tbl

f_info.tbl

f_match.tbl

f_stat.tbl

interfaces.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

fwlogtbl syntax

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

Example:

```
fwlogtbl -a  -d :c\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. Thus your Windows NT Firewall can process both firewall and su log files from earlier AIX Firewalls.

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 89.

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:

```
db2cmd
```

Then in the resulting DB2 command window enter:

```
db2 create database fwlog
```

You must then connect to the fwlog database:

```
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 fwqrysmp.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 fwqrysmp.dml files to place the creator ID in front of each table name.

The `ROOTDIR\sample\report\fwschema.ddl` file contains the DDL statements to create the database tables needed to accept records from the tabulated files created by **fwlogtbl**. *ROOTDIR* is the directory that you have selected during the installation process as the target location for the IBM Firewall. 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 ROOTDIR\sample\report\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 32, 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 ROOTDIR:\sample\report\fwqrysmp.dml file 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 32, 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, use the command line.

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
INTERFACES
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_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 Windows NT, and messages that are common to both firewalls.

For more information on firewall log messages, see "Appendix A. Messages" on page 89.

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, Windows NT thread ID (required)
MSG_NUM	int	Message number (required)

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

Column	Data Type	Short Description
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)
ERROR_NUM	small int	System Error number -- AIX errno or Windows NT Last Error (ICA0006, ICA0007, ICA0008, ICA0009, ICA0010, ICA0011, ICA0015)
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, Windows NT thread ID (required)

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

Column	Data Type	Short Description
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, Windows 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)

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

Column	Data Type	Short Description
FIREWALL	char(100)	Fully qualified name of the firewall machine (required)
PID	int	AIX Process ID, Windows 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)
PROTOCOL	char(7)	High-level protocol such as UDP, IPIP, ICMP, TCP or TCP / ACK (required)
SRC_PORT	int	• IP Packet type for ICMP • Resource protocol port number for others (required)
DST_PORT	int	• 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(10)	Encryption algorithm: 3DES_CBC, DES_CBC, 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)

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

Column	Data Type	Short Description
FIREWALL	char(100)	Fully qualified name of the firewall machine (required)
PID	int	AIX Process ID, Windows 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. INTERFACES. This table contains interface (adapter) configuration message information from the interface.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, Windows NT thread ID (required)
MSG_NUM	int	Message number (required)
IP	char(15)	IP address for the adapter (ICA9038, ICA9039, ICA9040)
OLD_MASK	char(15)	Previous mask value (ICA9040)
NEW_MASK	char(15)	New mask value (ICA9040)
Note: Related Messages: ICA9037, ICA9038, ICA9039, ICA9040, ICA9041		

| Table 7. MAIL_TRACK. This table contains mail tracking message information from the mail_trk.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, Windows NT thread ID (required)
MSG_NUM	smallint	Message number (required)

Table 7. MAIL_TRACK (continued). This table contains mail tracking message information from the mail_trk.tbl file.

Column	Data Type	Short Description
HOST	varchar(254)	Host name (ICA2251)
TRACK_ID	varchar(254)	Mail tracking identifier (ICA2251)
SUBJECT	varchar(254)	Mail subject (ICA2251)
MAIL_FROM	varchar(254)	Mail source (ICA2251)
MAIL_TO	varchar(254)	Mail destination (ICA2251)
MAIL_SIZE	int	Mail size (ICA2251)
ORIGIN_FROM	varchar(254)	Originator as received (ICA2251)
ORIGIN_TO	varchar(254)	Originator as transmitted (ICA2251)
DEST_FROM	varchar(254)	Destination as received (ICA2251)
DEST_TO	varchar(254)	Destination as transmitted (ICA2251)
STATUS	int	Status (ICA2251)
Note: Related Message: ICA2251		

Table 8. NAT_INFO. This table contains Network Address Translation message information from the nat_info.tbl file. If you have a report utilities database, the NAT_INFO table configuration must be altered. Issue the SQL command ″ALTER TABLE NAT_INFO ADD RC INTEGER;″.

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, Windows 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)
RC	int	Internal firewall return code (ICA9043)
Note: Related Messages: ICA9032, ICA9033, ICA9034, ICA9035, ICA9036, ICA9042, ICA9043, ICA9044, ICA9046)		

Table 9. 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)

Table 9. PAGER_INFO (continued). This table contains information related to the paging feature of the Firewall, from the pgr_info.tbl file.

Column	Data Type	Short Description
FIREWALL	char(100)	Fully qualified name of the firewall machine (required)
PID	int	AIX Process ID, Windows 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 10. 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, Windows NT thread ID (required)
MSG_NUM	int	Message number (required)
USERID	char(16)	User ID (required)

Table 10. PROXY_FTP (continued). This table contains FTP action information from FTP sessions from the p_ftp.tbl file.

Column	Data Type	Short Description
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 transfered
SID	long int	Unique session ID (required)
Note: Related Message: ICA2075		

Table 11. 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)
FIREWALL	char(100)	Fully qualified name of the firewall machine (required)
PID	int	AIX Process ID, Windows 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 transfered.
Note: Related Message: ICA2099		

Table 12. 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, Windows NT thread ID (required)
MSG_NUM	int	Message number (required)
USERID	char(16)	User ID (ICA2018, ICA2019, ICA2057, ICA2058, ICA2166, ICA2177, ICA2172)

Table 12. 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
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)
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)

Table 12. 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
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 13. 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, Windows 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 14. PROXY_STATUS. If you have an existing Report Utilities database, the PROXY_STATUS table configuration must be altered. Issue the SQL command \″ALTER TABLE PROXY_STATUS ADD SID INTEGER ADD SOCKET CHAR(25) ADD RC SMALLINT ADD CMD CHAR(25) ;\″. If you do not have an existing database, the change will be taken care of automatically when you create the database.) The updated PROXY_STATUS table description follows.

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)

Table 14. PROXY_STATUS (continued). If you have an existing Report Utilities database, the PROXY_STATUS table configuration must be altered. Issue the SQL command \″ALTER TABLE PROXY_STATUS ADD SID INTEGER ADD SOCKET CHAR(25) ADD RC SMALLINT ADD CMD CHAR(25) ;\″. If you do not have an existing database, the change will be taken care of automatically when you create the database.) The updated PROXY_STATUS table description follows.

Column	Data Type	Short Description
PID	int	AIX Process ID, Windows 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)
REMOTE_HOST	char(100)	Remote host name (from perspective of firewall machine) (ICA2021, ICA2022, ICA2027)
VERSION	char(10)	Proxy server version (ICA2097)
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 command (ICA2182)
Note: Related Messages: ICA2000 ICA2010 ICA2011 ICA2012 ICA2013 ICA2016 ICA2021 ICA2022 ICA2027 ICA2097 ICA2098 ICA2141 ICA2163 ICA2164 ICA2165 ICA2177 ICA2180 ICA2181 ICA2182		

Table 15. 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, Windows NT thread ID (required)
MSG_NUM	int	Message number (required)
USERID	char(16)	User ID (ICA9003, ICA9004)

Table 15. SERVER_INFO (continued). This table contains information about Configuration Server status and activities from the srv_info.tbl file.

Column	Data Type	Short Description
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 16. 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 Process ID, Windows 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)	• 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.
SID	long int	Unique session identifier, generated by the Firewall, based on clock time.

Table 16. SESSION (continued). This table contains SOCKS and PROXY session start/stop information from the session.tbl file.

Column	Data Type (length)	Short Description
Note: Related Messages: • 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) Details of Socks FTP session actions are in SOCKS_FTP table. Details of Proxy FTP session actions are in PROXY_FTP.		

Table 17. 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 Process ID, Windows 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)	• ’start’ when data bind is established.(ICA3010) • ’stop’ when data bind is terminated.(ICA3014) (required)
BYTES	long int	Amount of data transfered.
Note: Related Messages: ICA3010 ICA3014		

Table 18. 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, Windows 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)
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)

Table 18. 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
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		

Table 19. 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, Windows NT thread ID (required)
MSG_NUM	int	Message number (required)
Client_IP	char(15)	IP address of the client
Note: Related Messages: ICA5015 ICA5022 ICA5023 ICA5028 ICA5029 ICA5036 ICA5039 ICA5060 ICA5063 ICA5082 ICA5120		

Table 20. 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 21. 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, Windows 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(10)	Encryption algorithm: 3DES_CBC, DES_CBC, or CDMF
Note: Related Message: ICA1043		

Table 22. TUNNEL_STATUS. This table contains information on status changes of TUNNELS from the t_stat.tbl file. If you have the report utilities database, the tunnel_status configuration table must be altered. Issue the SQL command ″ALTER TABLE TUNNEL_STATUS ADD ACTIVATE TIMESTAMP DEACTIVATE TIMESTAMP;″.

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	long int	AIX Process ID, Windows 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 (for ICA1041, ICA1057, ICA1058, ICA1059, ICA1060 ICA6000, ICA6001)
RC	long int	Return code (ICA1053, ICA1054, ICA1055, ICA1057, ICA1058, ICA1059, ICA1060)
ACTIVATE	date_time	Date and time of the tunnel activation (ICA6000)
DEACTIVATE	date_time	Date and time of the tunnel deactivation (ICA6001)

Table 22. TUNNEL_STATUS (continued). This table contains information on status changes of TUNNELS from the t_stat.tbl file. If you have the report utilities database, the tunnel_status configuration table must be altered. Issue the SQL command ″ALTER TABLE TUNNEL_STATUS ADD ACTIVATE TIMESTAMP DEACTIVATE TIMESTAMP;″.

Column	Data Type	Short Description
Note: Related Messages: ICA1038, ICA1039, ICA1041, ICA1042, ICA1053, ICA1054, ICA1055, ICA1057, ICA1058, ICA1059, ICA1060, ICA6000, ICA6001 • 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. Log Archiver Plug-in Software Development Kit

The IBM Firewall log daemon writes logging information to the files that you specify with the **Log Facilities** dialog box of the configuration client. You then use the `fwlogmgmt` command to periodically archive old log records. Typically, you run the `fwlogmgmt` command from the Windows NT Scheduler. The `fwlogmgmt` command archives old log records into a directory and compresses them using the Windows NT `compact` command. However, you can write a Log Archiver plug-in to replace the default archive behavior.

How to Create a Log Archiver Plug-in

To create a Log Archiver plug-in you have to:

1. Write the source code for the plug-in DLL
2. Build the DLL
3. Install the DLL on the Firewall

The `ROOTDIR\sample\logarch` directory contains sample code for a log archiver plug-in that duplicates the Firewall's default behavior and a make file for IBM VisualAge for C++. *ROOTDIR* is the directory that you have selected during the installation process as the target location for the IBM Firewall.

Writing the Source Code

The Log Archiver plug-in must implement a set of functions that the Firewall uses to perform the archiving function. The prototypes for these functions are defined in fwarch.h in the `ROOTDIR\sample\logarch` directory.

These functions implement basic archiving functions like adding a file to an archive, extracting a file from an archive, refreshing an archive, and listing files in an archive.

See the sample code in fwarch.c in the `ROOTDIR\sample\logarch` directory for more details on these functions.

Building the DLL

When you have written the source code for the Log Archiver plug-in, you must compile and link it into a DLL. The DLL must be named fwarch.dll. All of the functions listed in fwarch.h must be exported from the DLL.

A sample make file for IBM VisualAge for C++ to build the sample code into the appropriate DLL is provided in the `ROOTDIR\sample\logarch` directory.

Installing the DLL

After you have successfully built the fwarch.dll, install it on the Firewall. Copy the fwarch.dll into the `ROOTDIR\bin` directory.

The Firewall’s default fwarch.dll is located in this directory also. Back up or rename this DLL before copying your replacement DLL into the directory.

Also, ensure that the `fwlogmgmt` command is not currently running and that the IBM Firewall log daemon is not running when you replace the default DLL. Use the Services Control Manager to stop the IBM Firewall log daemon and then restart it after you have replaced the DLL.

Chapter 4. Providing Your Own Authentication Methods

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

User-Supplied Authentication

We provide a user-authentication sample that is located in the directory `ROOT_DIR\bin\authsdk`. The files included are:

- authschm.h - interface definition files
- authus.cpp - source file for sample scheme
- gwauth4.lib - Firewall’s library
- msvc++.mak - Microsoft Visual C Make file
- schmname.h - interface definition files
- vac++.mak - IBM VisualAge Make file

Use the following commands to compile the user-authentication sample for IBM VisualAge:

- nmake -f vac++.mak - builds the DLL
- nmake -f vac++.mak install - builds and installs the DLL
- nmake -f vac++.mak clean - cleans up the local directory

Use the following commands to compile the user-authentication sample for Mircosoft Visual C:

- nmake -f msvc++.mak - builds the DLL
- nmake -f msvc++.mak install - builds and installs the DLL
- nmake -f msvc++.mak clean - cleans up the local directory

Using the Software Development Kit to Create a User-Supplied Authentication Scheme

The IBM Firewall provides a plug-in interface to enable the integration of third-party authentication security products. It does this by writing an authentication scheme .dll that plugs into the Firewall’s authentication scheme interface.

Overview of Firewall Authentication Processing

The following firewall services must authenticate users before allowing them access to firewall services:

- IBM Firewall Configuration Server
- IBM Firewall Proxy FTP Daemon
- IBM Firewall Proxy HTTP Daemon
- IBM Firewall Telnet Daemon
- IBM Firewall Socks Server

The Firewall provides the following authentication schemes:

Deny All
The user is always denied access to the service.

Permit All
The user is allowed access to the service without being challenged.

Firewall Password
The user is challenged for a password that is defined in the Firewall User database.

NT Logon Password
The user is challenged for his or her Windows NT Logon Password.

SecurID Card
The user is authenticated with the Security Dynamics SecurID security card.

The authentication scheme used can be defined on a per user and per service basis. For example, the Firewall can be configured so that when user, *John*, tries to log on to the IBM Firewall configuration server he is challenged for his Windows NT Logon Password. But when *John* wants to use the IBM Firewall Telnet Proxy, he is authenticated using his SecurID Card. Meanwhile, when user, *Mary*, tries to log on to the IBM Firewall Configuration Server, she is challenged for her Firewall Password. See the administration chapter of the *IBM eNetwork Firewall User's Guide* for more information on the Firewall-supplied authentication schemes and how to define them for each user.

In addition to the authentication schemes provided by the IBM Firewall, you can install up to three user-supplied authentication schemes. You can write these schemes to interact with your existing security infrastructure or you can obtain them from third-party security vendors to integrate their products with the Firewall.

Each authentication scheme in the Firewall, including the user-supplied authentication schemes, is represented by a DLL that implements the authentication scheme API. This API defines how the authentication scheme registers itself with the Firewall and how the Firewall passes authentication requests to it.

Creating a User-Supplied Authentication Scheme

Creating a user-supplied authentication scheme consists of:

- Writing the source code to implement the authentication scheme API
- Compiling and linking the source code into a DLL
- Installing the DLL on the Firewall

C-source header files and library files needed to create a user-supplied authentication scheme, as well as sample code and sample make files for Microsoft Visual C++ and IBM Visual Age for C++, can be found in `ROOTDIR\bin\authsdk`.

Writing the Source Code

All authentication schemes must do two things:

1. Register themselves with the Firewall
2. Implement the AuthSchmFn

Register with the Firewall: Before the Firewall services are started, the Firewall attempts to load every DLL it finds in the `\bin\authschm` subdirectory. As each DLL is loaded, its initialization routine must call a function in the Firewall named registerAuthSchm in order to register itself with the Firewall.

The registerAuthSchm function prototype is defined in the authschm.h header file. It takes a single parameter that is a pointer to an AuthSchmInfo structure, which is also defined in authschm.h. The AuthSchmInfo structure associates an authentication scheme name with the address of the appropriate AuthSchmFn that the Firewall should call in order to pass authentication requests to the authentication scheme.

User-supplied authentication schemes must use one of the following three names:

1. user
2. userauth2
3. userauth3

There are symbolic names defined for these names in the header file schmname.h. User-supplied authentication schemes should be designed to allow the end user to specify which of these three names are used, so that multiple user-supplied authentication schemes can be installed on the same Firewall without having to worry about two different schemes requiring the same name.

After the DLL initialization routine has successfully called register AuthSchm and returned to the caller, the DLL should be prepared to process authentication requests. For this reason, it might be necessary to do any scheme-specific initialization in the DLL initialization routine also.

Implement AuthSchmFn: Each authentication scheme DLL must implement a function called AuthSchmFn using the prototype defined in authschm.h. The AuthSchmFn function has one parameter, a pointer to an AuthReq structure. The AuthReq structure is a simple C structure that contains all the information pertaining to the current authentication request. AuthReq is defined in authschm.h. The AuthReq structure contains the name of the user being authenticated, the Firewall component/service requesting the authentication and other information about the request. For a complete list and explanation of the information in the AuthReq structure, see the comments on it in authschmh.

In addition to the user name and firewall component, there are three parameters in the AuthReq structure that are particularly important in implementing an authentication scheme:

gwaput

This is the address of a call back routine supplied by the Firewall, which the authentication scheme can use whenever it needs to send a message to the user. For example, if the authentication scheme needs to issue a prompt message to the user, it would call the entrypoint supplied in the gwaput parameter to do so. The gwaput call back function is prototyped by the AuthSchmPut typedef in authschm.h. See the comments on the AuthSchmPut typedef for a complete list of parameters that the AuthSchmFn must pass in on this call.

gwaget

This is the address of a call back routine supplied by the Firewall, which the authentication scheme can use whenever it needs to retrieve a response from the end user being authenticated. For example, if the authentication scheme needs to get a password from the user, it would call the entrypoint supplied in the gwaget parameter to do so. The gwaget callback function is prototyped by the AuthSchmGet typedef in authschm.h. See the comments on the AuthSchmGet typedef for a complete list of parameters that the AuthSchmFn must pass in on this call. One parameter that is particularly important is the echo parameter. The AuthSchmFn can use this parameter to indicate whether the user’s response should be echoed back to him or not.

opaque_data

The opaque_data field is used by the Firewall to correlate calls to the AuthSchmFn with calls to its call back routines. When calling either the gwaget or gwaput routines, the AthSchmFn should pass in the same opaque_data value as was passed in to it on the AuthReq structure.

Note that authentication schemes must be able to interact with all of the Firewall components. Some of the Firewall components can support multiple challenge/response dialogs with the end user. These components are called interactive Firewall components. Some Firewall components, due to the nature of their protocols, can only support a single challenge/response. These are called non-interactive Firewall components.

The user-supplied authentication scheme must be able to modify its behavior based upon which Firewall component is calling it, as indicated by the component field of the AuthReq structure. The valid values for the component field are defined in authschm.h. The current valid values for the component field are:

Table 23. Valid Values for the Component Field

Component Symbol from AuthSchm.h	**Firewall Component**	**Interactive/Non-interactive**
AUTHSCHM_UNKNOWN	New or unrecognized Firewall component	Assume it is interactive
AUTHSCHM_REMADMIN	Configuration Server	Interactive
AUTHSCHM_FTP	FTP Proxy	Non-interactive
AUTHSCHM_TELNET	Telent Proxy	Interactive
AUTHSCHM_HTTP	HTTP Proxy	Interactive
AUTHSCHM_SOCKS_PWD	Socks Server using password authentication	Non-interactive
AUTHSCHM_SOCKS_CRAM	Socks Server using CRAM authentication	Interactive
AUTHSCHM_REMIPSEC	Remote Client IPSEC server (Currently not available on Windows NT)	Interactive

When the AuthSchmFn has completed its processing, it must return to the caller with one of the GWA return codes defined in authschm.h. This return code is used to indicate whether the user was successfully authenticated and whether or not there was an error during processing:

Table 24. GWA Return Codes

Return Code	Meaning
GWA_OK	No errors during processing and the user was successfully authenticated
GWA_DENY	No errors during processing, but the user failed to authenticate himself
GWA_IOFAILURE	An error occurred while trying to send prompts to the user or trying to get a response from the user. Typically this is returned when there are errors in the call back routines.
GWA_BUFFERTOOSMALL	The AuthSchmFn function was unable to retrieve a response from the user because it could not allocate a buffer big enough to receive the response.
GWA_NOAUTHFN	Error - Not relevant to authentication schemes
GWA_FNNOTREG	Error - Not relevant to authentication schemes
GWA_RSVNAME	Error- Authentication request contained a name that is reserved and cannot be used for this authentication scheme
GWA_BADNETTYPE	Error - Not relevant to authentication schemes
GWA_BADAPP	Error - Not relevant to authentication schemes
GWA_BADADDR	Error - Address supplied on authentication request was invalid
GWA_MEMSHORTAGE	Error - Authentication request could not be processed because memory could not be allocated
GWA_USERDBFAIL	Error - Could not query a required dababase
GWA_REGFAILED	Error - Not relevant to authentication schemes
GWA_AUTHERROR	Error - Authentication scheme specific error condition
GWA_INTERNAL	Error - Miscellaneous error condition in authentication scheme

When the AuthSchmFn returns to the Firewall, if the return code is GWA_OK, the user is considered to be authenticated and is given access to the requested service. GWA_DENY is treated as a non-error condition, but the user is denied access to the requested service. All other return codes are error conditions and the user is denied access to the requested service.

Compiling and Linking to the Source Code: When compiling and linking the source code into a DLL, you must link the DLL to gwauth4.dll using the gwauth4.lib supplied in the `\bin\authsdk` directory in order to resolve the entrypoint names defined in authschm.h. Also, it is important that the AuthSchmFn is not exported from the DLL. Sample make files for IBM VisualAge for C++ and Microsoft Visual C++ are supplied in the `\bin\authsdk` directory.

Installing the DLL: Once the DLL has been successfully built, copy it to the `ROOTDIR\bin\authschm` directory and reboot the Firewall machine. Rebooting is necessary in order for the Firewall to attempt to load the DLL and register the DLL's authentication schemes.

Putting it All Together: Figure 1 on page 62 shows how the authentication schemes are loaded and shows the key function calls during authentication request processing.

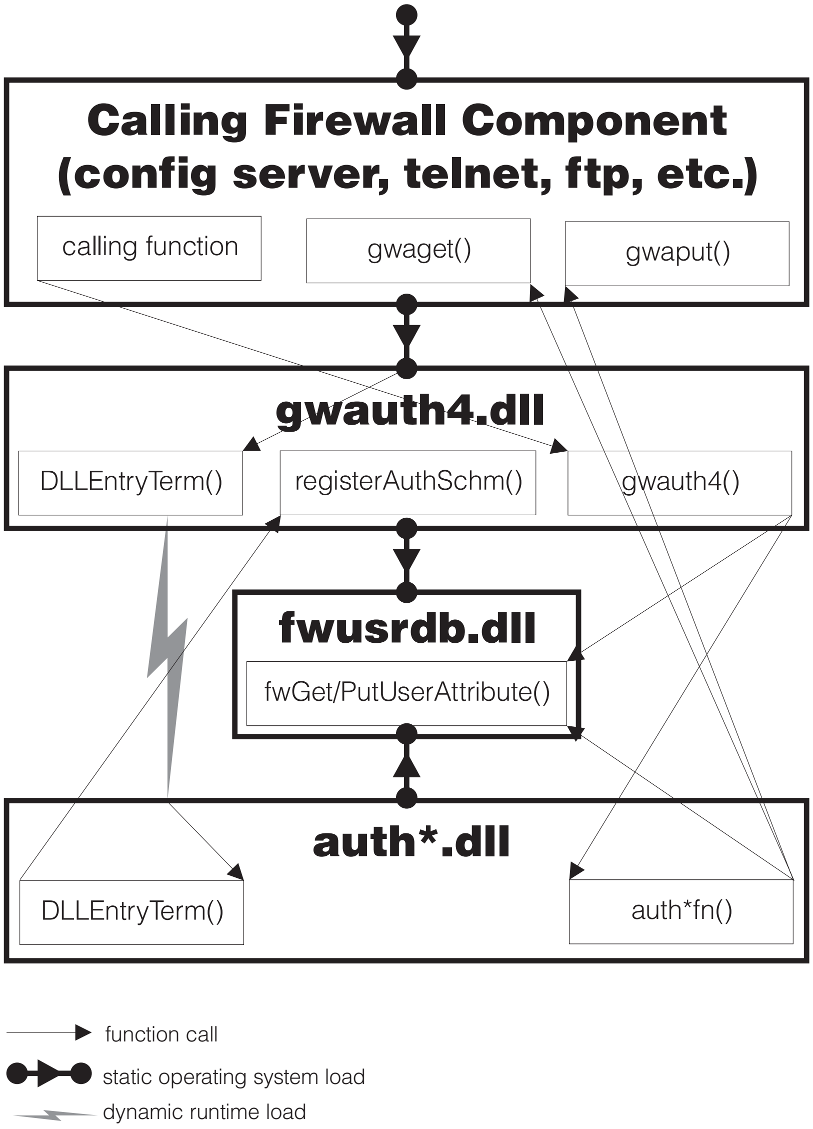

Figure 1. DLL Initialization and Registration

Firewall components that need to use the authentication services link to a Firewall DLL called gwauth4. When the gwauth4 dll is loaded, its DLLEntryTerm routine is invoked and will attempt a run time load of all DLLs in `ROOTDIR\bin\authschm`. If an authentication scheme DLL fails to load, it will not be considered an error for gwauth4 dll loading. The gwauth4 dll serializes these load attempts.

When the authentication schemes' DLLEntryTerm routine is run, they are responsible for registering the authentication schemes with the gwauth4.dll. This is done by calling registerAuthSchm. The authschm dll needs to call registerAuthSchm once for every authentication scheme that the DLL supports. The AuthSchmInfo structure that is passed in on the registerAuthSchm function associates the name of the authentication scheme as stored in the user database with the entry point of the AuthSchmFn function. The registration function will make copies of the structure passed into it, so that authschm dll can reuse/modify this structure as needed. The Authentication scheme DLL is also responsible for freeing the AuthSchmInfo structure.

The registerAuthSchm function is responsible for building a linked list that represents all of the registered authentication schemes. gwauth4's DLLEntryTerm routine will initialize the list anchor to NULL. Then when the authschm DLLs call the registerAuthSchm function it will:

1. Scan the authentication scheme list looking for an entry that has the same name as the name passed in. If one exists, remove it from the list and delete all of its associated storage.
2. Build an AuthSchmEntry structure based on the AuthSchmInfo structure and add it to the authentication scheme list.
3. Return to the caller an indication of whether the registration succeeded (GWA_OK) or failed (GWA_REGFAILED).

After gwauth4's DLLEntryTerm has performed a run time load on each of the authschm dlls and the authschm DLLs have registered their authentication schemes, gwauth4's DLLEntryTerm routine will return to the caller. At this point other components can begin requesting authentication services by calling the gwauth4 function.

When gwauth4.dll is unloaded, the DLLEntryTerm routine will be called again for termination processing. When called for termination, this routine will delete all AuthSchmEntry items on the AuthSchmList and their associated storage. This is done so that the authentication schemes do not have to deregister themselves from the Firewall.

Authentication Request Processing: When a Firewall service needs to authenticate a user, it calls functions in gwauth4.dll. gwauth4 takes information from that calling component and queries the Firewall user database to determine the name of which authentication scheme to use to process the request.

Once gwauth4 has determined the name of the authentication scheme, it scans its list of registered authentication schemes for a scheme by the same name. If it finds a registered scheme by the same name, it builds an AuthReq structure to represent the current request and calls the entrypoint in the authentication scheme DLL that is associated with the name.

The AuthSchmFn function called by gwauth4 processes the request and calls the gwaget and gwaput callbacks as needed to interact with the end user. When it completes its processing, it returns control to gwauth4 with an appropriate return code.

gwauth4 writes the appropriate log records to document the authentication request and then returns back to the firewall component that originated the request, propagating the return code that it received from the Authentication scheme DLL.

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

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 must be logged on using a Windows NT administrator account when running this utility.

1. Go to the ROOTDIR\config directory and start the key utility by entering:

   ```
   c:\program files\IBM\Firewall\config > 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 file name, 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 - PEM Certificate Request Format (Private Enhanced Message)
P - PKCS10 Certificate Request Format
C - Cancel

Enter a command: s
```

Enter ’s’, as shown above, to create a PEM Certificate Request Format. 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 1024.

  ```
  1:   508
  2:   512
  3:   768
  4:   896
  5:   1024
  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 creating the key file, you must specify the key file name in the configuration server parameter file using the fwcfgsrv command.

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

   After using the fwcfgsrv command, stop and restart the server service.

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.

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 6 for more information.

Installation and Setup

Filter support fails

Problem Explanation
You receive these error messages.

```
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.

DNS fails at installation time

Problem Explanation
You did not install the software in the prescribed order.

Recommended Action
Refer back to the Installation instructions and

1. Remove Microsoft DNS by deleting the entire directory: `\winnt\system32\DNS`
2. Reinstall Microsoft DNS
3. Reboot
4. Reinstall the DNS hotfix
5. Reboot

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 76 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 2. 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.

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 fails

Problem Explanation
You received DNS error messages because you configured Microsoft DNS Service with the Microsoft DNS Service Manager.

Recommended Action
Refer back to the Installation instructions and

1. Remove Microsoft DNS by deleting the entire directory: `\winnt\system32\DNS`
2. Reinstall Microsoft DNS
3. Reboot
4. Reinstall the DNS hotfix
5. Reboot

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.

You Have Used Microsoft Domain Name Service Manager to Configure DNS

Problem Explanation

DNS will not work on the Firewall. Microsoft DNS Manager has set DNS to boot from the registry and so the Firewall can no longer interpret DNS information.

Recommended Action

Click System Administration from the configuration client navigation tree. Click Domain Name Services and follow the instructions. You will get a message which will instruct you how to resolve the problem.

DNS Queries Fail or Time Out

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
Microsoft DNS service might not be running. Go to the service control manager to determine if it is running.

Recommended Action
Use the service control manager to start DNS.

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
Configuration client and configuration server are using different languages.

Recommended Action
On the configuration client log on panel, select the language in which the firewall has been installed.

Problem Explanation
SSL encryption may not be configured properly.

Recommended Action
Ensure that SSL is selected in the client's logon panel. Stop and restart the firewall configuration server using the service control manager.

Problem Explanation
The firewall's configuration server may be disabled.

Recommended Action
Ensure that the firewall configuration server is running.

Problem Explanation

The firewall's configuration server may be monitoring a non-standard port.

Recommended Action

Examine `c:\winnt\system32\drivers\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. Stop and restart the configuration server using the service control manager.

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 `fwcfgsrv` 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.

Traffic Control

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 76 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.

Authentication Services

A Windows NT administrator account cannot be authenticated

Problem Explanation

The firewall attributes for a Windows NT administrator account are stored in the firewall user database under `fwdfadm`.

Recommended Action

Verify that `fwdfadm` has the correct authentication method set for the service you are trying to use.

Firewall proxy user cannot be authenticated

Problem Explanation

If the firewall proxy user is not defined in the firewall user database, the `fwdfuser` name is used to define the user's attributes.

Recommended Action

Verify that `fwdfuser's` authentication method is defined correctly for the service that the user is trying to access.

Network Address Translation

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

When using NAT, IP packets traveling from the secure side of the Firewall to the nonsecure side have their source IP address and port number translated from the secure host's information to the NAT-configured IP address and port number. This means that the packets traveling back through the Firewall to the secure host need some help finding the Firewall host. For example, if a secure-side user initiates an FTP to a nonsecure host, those FTP packets (control connection and data connection) will have their source IP address and port changed on the way outbound to the FTP server. As it responds to those FTP packets, the FTP server is going to use the source information as its response packet's destination IP address and port. Unfortunately, network routing between the FTP server and the Firewall host (behind which the secure host FTP client awaits the Server's response) does not know where to route IP packets with the NAT-configured IP address for the destination.

If a Firewall administrator has access to the nonsecure router's routing table a static route can be added to those tables telling the router exactly where to send NAT-translated packets. However, often the Firewall administrator either does not have such access to the router's tables, it is inconvenient to update those route tables with a static route, or the Firewall administrator does not even know where the router resides if it is not under his or her control.

To solve this problem, IP aliasing can be used to associate multiple IP addresses with an interface. To enable this on Windows NT:

1. Select:

   ```
   Settings
     Control Panel
       Network
        Protocols
          TCP/IP Protocol
            Properties
              IP Address (tab)
   ```

2. Highlight the host's nonsecure adapter in the **Adapter** window, and click the **Advanced...** box.
3. In the **IP Address** box, click **Add...** and then enter the NAT-configured IP address and subnet mask. Note Windows NT limits the number of IP aliases per interface.
4. Close and reboot.

Due to the way routers identify interfaces on which to route a packet, it is important to have a NAT-configured IP address that is in the same subnet as the Firewall's nonsecure IP address.

Why Can't I PING the NAT many-to-one address or the NAT MAP address?

Problem Explanation

NAT does not support translation of ICMP packets (Query type or Error type). Because NAT does not support ICMP, NAT will discard packets that either will be impossible to route correctly (for example, a PING of the many-to-one address), or might represent a security exposure (for example, PING of a nonsecure host from a to-be-NAT translated secure host).

Recommended Action

If you want to test your network's routing through the Firewall using programs like PING or TRACERT, do so before activating NAT. After activating NAT, do not attempt to PING or TRACERT the NAT many-to-one or MAP addresses as this is not supported.

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.

NAT Configuration Statements

The following table summarizes NAT's behavior for various combinations of configuration rules. The configuration client or the command line interface (fwnat cmd=list) will show the current NAT configuration. If for example, the output of an `fwnat cmd=list` shows:

```
 1. MANY-TO-ONE  204.114.22.3  255.255.255.255  15
 2. MAP  10.1.1.1   9.37.51.211
 3. MAP  10.1.1.2   9.37.51.212
```

Then the corresponding NAT behavior description in the table is in the row that has an Active NAT Entry of **MANY-TO-ONE and MAP**.

Active NAT Entry	**What NAT Code Does**
None	When there are no active NAT configuration file entries, NAT is not active and no secure addresses are translated.
MANY-TO-ONE only	All secure source addresses are translated in all outbound packets.
TRANSLATE only	Outbound packets with secure source addresses matching the TRANSLATE entry are discarded because there is no MANY-TO-ONE entry specifying available external addresses. Outbound packets that don't match the TRANSLATE are allowed through without translation.
EXCLUDE only	Outbound packets with secure source addresses matching the EXCLUDE entry are allowed through without translation. Outbound packets that do not match the EXCLUDE are discarded.

Active NAT Entry	What NAT Code Does
MAP only	Packets with secure source or destination addresses matching the MAP entry are translated. Outbound packets that do not match the MAP are allowed through without translation.
MANY-TO-ONE and TRANSLATE	Outbound packets with secure source addresses matching the TRANSLATE entry are translated because there is a MANY-TO-ONE entry specifying available external addresses. Outbound packets that do not match the TRANSLATE are allowed through without translation.
MANY-TO-ONE and EXCLUDE	Outbound packets with secure source addresses matching the EXCLUDE entry are allowed through without translation. Outbound packets that do not match the EXCLUDE are translated.
MANY-TO-ONE and MAP	Packets with secure source or destination addresses matching the MAP entry are translated. Outbound packets that do not match the MAP entry are translated.
EXCLUDE and TRANSLATE	Outbound packets with secure source addresses matching the EXCLUDE entry are allowed through without translation. Outbound packets with secure source addresses matching the TRANSLATE entry are discarded because there is no MANY-TO-ONE entry specifying available external addresses. Outbound packets that do not match either entry are allowed through without translation even though they do not match the EXCLUDE entry.
EXCLUDE and MAP	Outbound packets with secure source addresses matching the EXCLUDE entry are allowed through without translation. Packets with secure source or destination addresses matching the MAP entry are translated. Outbound packets that do not match either entry are allowed through without translation.

Active NAT Entry	What NAT Code Does
MAP and TRANSLATE	Packets with secure source or destination addresses matching the MAP entry are translated. Outbound packets with secure source addresses matching the TRANSLATE entry are discarded because there is no MANY-TO-ONE entry specifying available external addresses. Outbound packets that do not match either entry are allowed through without translation.
MANY-TO-ONE, TRANSLATE, and EXCLUDE	Outbound packets with secure source addresses matching the TRANSLATE entry are translated because there is a MANY-TO-ONE entry specifying available external addresses. Outbound packets with secure source addresses matching the EXCLUDE are allowed through without translation. Outbound packets that do not match either entry are allowed through without translation even though they do not match the EXCLUDE entry.
MANY-TO-ONE, TRANSLATE, and MAP	Outbound packets with secure source addresses matching the TRANSLATE entry are translated because there is a MANY-TO-ONE entry specifying available external addresses. Packets with secure source or destination addresses matching the MAP are translated. Outbound packets that do not match either entry are allowed through without translation.
EXCLUDE, TRANSLATE, and MAP	Outbound packets with secure source addresses matching the EXCLUDE entry are allowed through without translation. Outbound packets with secure source addresses matching the TRANSLATE are discarded because there is no MANY-TO-ONE entry specifying available external addresses. Packets with secure source or destination addresses matching the MAP are translated. Outbound packets that do not match any entry are allowed through without translation.

Active NAT Entry	What NAT Code Does
MANY-TO-ONE, EXCLUDE, and MAP	Outbound packets with secure source addresses matching the EXCLUDE entry are allowed through without translation. Packets with secure source or destination addresses matching the MAP are translated. Outbound packets that do not match either entry are translated.
MANY-TO-ONE, TRANSLATE, EXCLUDE, and MAP	Outbound packets with secure source addresses matching the TRANSLATE entry are translated because there is a MANY-TO-ONE entry specifying available external addresses. Outbound packets with secure source addresses matching the EXCLUDE are allowed through without translation. Packets with secure source or destination addresses matching the MAP are translated. Outbound packets that do not match any entry are allowed through without translation.

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

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.dll > schema.out
db2 -vf fwimport.dat > import.out
db2 -vf fwqrysmp.dml > sample.out
```

Recommended Action

Provide the correct fully qualified filenames for the .dll, .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.

VPN Tunnels

The tunnel does not work

Problem Explanation

The most probable cause of a non-working tunnel are field entries that do not match the values entered on the partner system. Remember that your source addresses, SPIs and keys are reflected in the corresponding destination fields of your partner's tunnel defintion.

Recommended Action

Some products require a hexadecimal value for the SPI while others expect to receive a decimal number. In this case you have to manually convert the numbers before comparing them to your partner's definition. Of course, both tunnel partners need to use the same protocols, header formats, transforms and policies.

Nested tunnels are not working

Problem Explanation

Nested tunnels are not working.

Recommended Action

Try to get the tunnel between the firewalls to work. If that tunnel works, then give the nested tunnels a try.

Features that may not work with VPN tunnels

Problem Explanation

There are some features present in most available TCP/IP implementations that are useful for various purposes but may not work in conjunction with VPN tunnels.

Recommended Action

When TCP calculates the maximum segment size (MSS) it doesn't know about the extra length needed for IPSec headers. MTU path discovery turns on the 'Do not fragment' bit in the IP header in the TCP layer. Then the IP layer calls the IPSec kernel and adds the IPSec headers. If this added length exceeds an MTU, it will be rejected because of the 'Do not fragment' bit set. So TCP will retry after reducing the MSS, but it uses the MTU returned in the ICMP response and does not take into account the IPSec header length, so the retry will not work.

When IP datagrams carrying IPSec traffic are fragmented along the way, the general rule is to apply IPSec before fragmentation for outbound datagrams, and to apply IPSec after reassembly for inbound datagrams. This must be done in order to assure proper IPSec processing for authentication and/or encryption/decryption.

See the current Internet Drafts for IPSec, AH and ESP on the issues of fragmentation and path MTU discovery.

AIX host-firewall-host tunnels do not import into Windows NT

Problem Explanation

For host-host tunnels and host-firewall-host tunnels, the AIX operating system will default the destination policy to auth/encr for an authentication with AH and encryption with ESP tunnel. So when the tunnel is imported to Window NT, it fails because Windows NT does not support auth/encr. This is a problem because dynamic filter tunnels using local and remote user addresses do not import into AIX.

Recommended Action

Create tunnels on Windows NT and export them to AIX. However, if you choose not to do this go to the SMIT *Change a Manual Tunnel* panel and switch the destination policy back to encr/auth because the AIX operating system will have defaulted it to auth/encr.

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 73.

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

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 alloweded.

Explanation: Multiple entries for a log file in the logmgmt.cfg is not alloweded.

ICA0054 Can not open *file name* file.

Explanation: Unable to open the named 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,″*message text*″, is invalid

Explanation: The text shown is not a valid log message.

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 denial-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 = 0x***function 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/ipsp_poif***:** *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 directionl= log_control f=fragment_controlt= 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 *number*

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 *number*

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.

ICA1055 SCCS internal error: error type is *number*

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 *number*

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?

ICA1057 **Resource allocation failure for tunnel-id:** *tunnel-id*: **error type is** *number*

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.

ICA1058 **Communication error for tunnel-id:** *tunnel-id*: **error type is** *number*

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.

ICA1059 **SCCS internal error for tunnel-id:** *tunnel-id*: **error type is** *number*

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.

ICA1060 **Timeout failure for tunnel-id:** *tunnel-id*: **error type is** *number*

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?

ICA1061 **TC_LOG***number*: **Tunnel interface module for IPv***number* **was started at** *time* **on** *date*

Explanation: IPSEC: Tunnel interface module was started.

ICA1062 **TC_LOG***number*: **Tunnel interface module for IPv***number* **was shutdown at** *time* **on** *date*

Explanation: IPSEC: Tunnel interface module was shutdown.

ICA1063 **TC_LOG***number*: **Tunnel cache module for IPv***number* **was started at** *time* **on** *date*

Explanation: IPSEC: Tunnel cache module was started.

ICA1064 **TC_LOG***number*: **Tunnel cache module for IPv***number* **was shutdown at** *time* **on** *date*

Explanation: IPSEC: Tunnel cache module was shutdown.

ICA1065 **TC_LOG***number*: **Tunnel** *number* **with ESP SPI** *number* **and AH SPI** *number* **for IPv***number* **was activated at** *time* **on** *date*

Explanation: IPSEC: tunnel is activated

System Action: Establish a tunnel.

ICA1066 **TC_LOG***number*: **Tunnel** *number* **with ESP SPI** *number* **and AH SPI** *number* **for IPv***number* **was deactivated at** *time* **on** *date*

Explanation: IPSEC: tunnel is deactivated

System Action: Close a tunnel.

ICA1067 **TC_LOG***number*: **Tunnel cache for IPv***number* **was cleared at** *time* **on** *date*

Explanation: IPSEC: a tunnel is cleared from tunnel cache

System Action: Tunnel cache is updated

ICA1068 **TC_LOG***number*: **Tunnel** *number* **not found at** *time* **on** *date*

Explanation: IPSEC: tunnel is not available

ICA1069 **TC_LOG***number*: **Tunnel cache entry not found. src addr=***IP-address***,dst addr=***IP-address***,SPI=***number***,tunnel id=***number* **at** *time* **on** *date*

Explanation: IPSEC: tunnel entry is not available in cache

ICA1070 **TC_LOG***number*: **AH failure. src addr=***IP-address***,dst addr=***IP-address***,SPI=***number***,flow id=***number* **at** *time* **on** *date*

Explanation: IPSEC: AH failure.

ICA1071 **TC_LOG***number*: **ESP failure. src addr=***IP-address***,dst addr=***IP-address***,SPI=***number***,flow id=***number* **at** *time* **on** *date*

Explanation: IPSEC: ESP failure.

ICA1072 **TC_LOG***number*: **Tunnel expired. src addr=***IP-address***,dst addr=***IP-address***,ESP SPI=***number***,AH SPI=***number***,tunnel id=***number* **at** *time* **on** *date***;**

Explanation: IPSEC: Tunnel expired.

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 protocol specification.**

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 = 0x***function 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 0x*hex_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 exists, 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 ptelnet 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 n 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. The most likely cause of this error is another ftp daemon already listening on the standard ftp port (21).

User Response: Correct the indicated system problem and restart pftpd.

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 availible; use Firewall commands/smit panels

ICA2089 Authorization type from user file (*Authorization type*) did not match any entries in table (struct tab2 authtab[]).

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 invaild 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 invaild 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 occured. The host Login Monitor service, lm, is not specified properly in the /etc/services file.

ICA2111 **byname() error**: *errno*.

Explanation: An gethostbyname() error occured. 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 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 coverting 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 Connectting 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.**

Explanation: The indicated function successfully authenticated the indicated user id.

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 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 n defined any user to use User-Supplied authentication. The IBM-supplied n fwuser.o denies access to all non-AIX and non-Firewall users.

ICA2147 fwtelnet: 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 (fwtelnet).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 fwtelnet: a LOGIN_ADAPTER_ERROR occured while starting a transparent telnet session from *source IP addr* **to** *dest IP addr***.**

Explanation: A LOGIN_ADAPTER_ERROR occured when calling q_check_secure(0).

System Action: Reject the connection.

User Response: check the secure adapter.

ICA2150 Pftpd error - *failing function***: return code = 0x***function 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 fwtpproxy 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.

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: (1) Incorrect ″old″ password was specified, (2) Only one occurrence of the ″new″ password was specified, (3) Two occurrences of ″new″ password do not match, or (4) 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 **safemaild started.**

Explanation: Starting safemaild.

ICA2164 **safemaild stop.**

Explanation: stopping safemaild.

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 specifed 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 registerd 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.

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 specifed 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 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 0x***session 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 0x***session 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 0x***session 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 0x***Session 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 0x***Session 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 0x*Session 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.*. safemaild exit.

Explanation: System error when safemaild daemon attempted to establish signal handler.

ICA2185 Cannot open socket. safemaild exit

Explanation: Failure while opening the socket.

ICA2186 Cannot bind the socket to the port. safemaild 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. safemaild 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 occured 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 occured, 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 begining 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., ptelnetd) 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 begining 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., ptelnetd) 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

ICA2251 **hostname =** *Host Name***, msg_id =** *Message ID***, subject =** *Subject***, input_mta =** *Input MTA***, output_mta =** *Output MTA***, msg_size =** *Message Size***, orig_in =** *Originator (as we received it)***, orig_out =** *Originator (as we transmitted it)***, recip_in =** *Recipient (as we received it)***, recip_out =** *Recipient (as we sent it)***, sent =** *Sent Status***.**

Explanation: The tracking records for all the connections attempted to estabilish.

ICA2252 **Could not set socket options %1$s**

Explanation: An attempt to configure the socket has been rejected by the operating system for the reason listed in the message. The configuration of the TCP/IP stack and operating system security should be verified.

ICA2253 **Could not map socket to streams: %1$s**

Explanation: An attempt to map the descriptors for the socket to a stream has been rejected by the operating system for the reason listed in the message. The configuration of the TCP/IP stack and operating system security should be verified.

ICA2254 **Could not open socket: %1$s**

Explanation: An attempt to open a socket has been rejected by the operating system for the reason listed in the message. The configuration of the TCP/IP stack as to the availability and permissions for the socket specified in the configuration should be verified.

ICA2255 **Could not bind to socket: %1$s**

Explanation: An attempt to bind to a socket has been rejected by the operating system for the reason listed in the message. The configuration of the TCP/IP stack as to the availability and permissions for the socket specified in the configuration should be verified.

ICA2256 **Could not listen to socket: %1$s**

Explanation: An attempt to listen to a socket has been rejected by the operating system for the reason listed in the message. The configuration of the TCP/IP stack as to the availability and permissions for the socket specified in the configuration should be verified.

ICA2257 **Could not enable nonblocking mode for socket: %1$s**

Explanation: An attempt to enable nonblocking mode for a socket has been rejected by the operating system for the reason listed in the message. The configuration of the TCP/IP stack as to the availability and permissions for the socket specified in the configuration should be verified.

ICA2258 Serious I/O error getting the socket name: %1$s

Explanation: An attempt to get name for a socket has been rejected by the operating system for the reason listed in the message. The configuration of the TCP/IP stack as to the availability and permissions for the socket specified in the configuration should be verified.

ICA2259 Error reading from %1$s: %2$s

Explanation: An attempt to read from a connection with the hostname listed has failed for the listed reason. This could occur for instance because the TCP/IP connection has been disrupted or an error occurred in the device or application that was connected.

ICA2260 No parsable MX records found for %s

Explanation: An attempt to use DNS to route a message to a hostname using DNS has resulted in no MX records. Check the DNS configuration for the hostname listed to correct the problem.

ICA2261 Error writing temporary file: %s

Explanation: While trying to write recovery information to a temporary file the operating system reported the listed error. Correct the problem in the operating system or environment.

ICA2262 I/O error writing to temporary storage: %s

Explanation: While trying to write recovery information to a temporary file the operating system reported the listed error. Correct the problem in the operating system or environment.

ICA2263 Error sending body part: %s

Explanation: While trying to send the body of a message in the data segment of the SMTP transfer the listed error occurred. This could be the result of a problem in the system to which the connection was made or in the TCP/IP stack for instance. If possible verify the proper operation of the receiving system.

ICA2264 I/O error sending body part: %s

Explanation: While trying to send the body of a message in the data segment of the SMTP transfer the listed error occurred. This could be the result of a problem in the system to which the connection was made or in the TCP/IP stack for instance. If possible verify the proper operation of the receiving system.

ICA2265 Failed to send fallback message: %s

Explanation: AThe attempt to send a message to the fallback system has failed and the message was not able to be delivered This could be the result of a problem in the system to which the connection was made or in the TCP/IP stack for instance. If possible verify the proper operation of the fallback system.

ICA2266 Conversation failure: %1$s : %2$s

Explanation: An I/O error has occurred in the module name listed with the error listed. This is most likely the result of a problem with the operating environment.

ICA2267 Cannot start a processing thread : will shutdown: %s

Explanation: An attempt to start a processing thread by the main processing task has failed and processing is being shutdown. Correct the problem with the operating environment and retry.

ICA2268 Fatal error: unable to wait on mother board semaphore, error code (%s)

Explanation: An attempt to wait on a sempahore by the main processing task has failed and processing is being shutdown. Correct the problem with the operating environment and retry.

ICA2269 Connection with the overflow host failed: %s

Explanation: The attempt to connect to the fallback system has failed with the listed error code. This could be the result of a problem in the system to which the connection was made or in the TCP/IP stack for instance. Verify the proper operation of the fallback system.

ICA2270 Overflow host refused fallback mail: %s

Explanation: The attempt to send a message to the fallback system has failed with the listed error code. Correct the configuration or operating environment for the fallback system.

ICA2271 Overflow MTA has failed.

Explanation: The attempt to send a message to the fallback system has failed. Correct the configuration or operating environment for the fallback system.

ICA2272 Overflow rejected recipient: %1$s for (%2$s)

Explanation: The attempt to send a message to the fallback system has failed. The actual address being used to deliver the message is listed as well as the preferred mail address. Correct the configuration or operating environment for the fallback system.

ICA2273 Could not open the overflow channel for %1$s(%2$s)

Explanation: The attempt to send a message to the fallback system has failed. The actual address being used to deliver the message is listed as well as the preferred mail address. Correct the configuration or operating environment for the fallback system.

ICA2274 Shutdown signal received

Explanation: A signal to shutdown has been received and processing will stop.

ICA2275 Listening for Connections.

Explanation: Initialization has completed successfully.

ICA2276 Received a connection from: %s

Explanation: A connection has been established by the system with the hostname listed.

ICA2277 Connection closed with: %s

Explanation: A connection has been closed with the system listed.

ICA2278 Monitor is Starting.

Explanation: The main processing task that monitors the operation of the processing threads has started.

ICA2279 Monitor is Terminating.

Explanation: The main processing task that monitors the operation of the processing threads is terminating and processing will cease.

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: Shell command displayed between < and > 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 (0x***hex-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 (0x***hex-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 Sockd could not fork daemon: *system error message*

Explanation: (SOCKS422) Fork to initialize sockd daemon failed.

User Response: Correct the indicated system problem and restart sockd.

ICA3123 Sockd server starting.

Explanation: (SOCKS422) Sockd 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.

ICA3136 **Sockd error -** *failing function***:** *error message*

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 occured 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 **IpcKey: 0x***IpcKey*

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 *pagerNumber* **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 *pagerNumber* **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 *pagerNumber*

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 paramters 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: 0x*terminator*

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: 0x***IpcKey***.**

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 Request .....OK!

Explanation: Got [p back, meaning that the carrier received the AMR.

ICA4310 Sending out message

Explanation: Sending out message block over

ICA4311 Waiting for result

Explanation: Waiting for the confirmation

ICA4312 Ack received. Page successful

Explanation: The Ack control character was received from the carrier, indicating that the page was sent successfully.

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 to resolve the problem if it persists.

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 bits 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 versin. 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 page 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.

ICA6000 Tunnel *tunnel id* **was successfully activated at:** *time***.**

Explanation: The tunnel was successfully activated at the given time.

ICA6001 Tunnel *tunnel id* **was successfully deactivated at:** *time***.**

Explanation: The tunnel was successfully deactivated at the given time.

ICA8001 SYSLOG/udp: unknown service

Explanation: Processing terminated because the syslog service was unknown or unavailable.

User Response: Contact the system programmer. SystemProgrammer :Check the /etc/services or tcpip.ETC.SERVICES data sets for the existence of syslog.

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

Explanation: This message is generated in addition to a select message.

User Response: None. SystemProgrammer :None.

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.

ICA8009 SYSLOGD exiting on signal *signal*

Explanation: The syslog daemon received a signal that has caused the syslog daemon to exit.

ICA8010 SYSLOGD restarted

Explanation: The syslog daemon has been 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 = 0x*return_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.

ICA8017 *function_name* **function failed -** *error text*

Explanation: Processing terminates because syslogd could not perform the specified function. The error text is appended to the error message.

User Response: Contact the system programmer. SystemProgrammer :Use the error text to determine the cause of the failure.

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 = 0x***return_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

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.

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.

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.

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.

ICA9041 No interfaces were found on this machine.

Explanation: No adapter interfaces were found on this machine.

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.

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.

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.

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.

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 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 dategram.

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 ICA9050e is issued, some severe event is happening or has happened on the Firewall host that is causing NAT to fail and the packet to be dropped. That severe event could be, for example: 1) A normal or abnormal shutdown of the Firewall host has occurred. 2) The Kernel is currently severely short of memory. 3) An internal NAT processing error has occurred. There is no recommended user response to error log message ICA9050e other than to contact IBM service and report the problem if knowledge of the state of the Firewall host and any possible severe events it has undergone doesn't already explain it. Supplying the return code will help IBM service pinpoint the nature of the severe event and where in the NAT code the severe event was detected. The ICA9050e error log message is issued to make the user aware that packets are being dropped by the Firewall.

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.

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.

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

ICA9056 **NAT could not translate a packet, protocol ICMP (type** *type***, code** *code***), source address** *source address***, destination address** *destination address***.**

Explanation: Unlike NAT error log message ICA9050e, this NAT informational log message indicates NAT's inability to translate a packet due to a current NAT functional limitation. Although NAT cannot perform the translation, NAT's configuration data indicates these packets should not pass through the Firewall without NAT translation. Rather than risk a security exposure by allowing untranslated packets to flow through to the non-secure network, NAT drops the packet. This log message is informational because it does not indicate an error has occurred.

System Action: The packet is dropped.

User Response: none

ICA9060 **Fatal configuration server initialization error - socket():** *system error message*

Explanation: Configuration server initialization failed, daemon ended.

User Response: Correct the specified system problem and start the configuration server again.

ICA9061 **Fatal configuration server initialization error - listen():** *system error message*

Explanation: Configuration server initialization failed, daemon ended.

User Response: Correct the specified system problem and start the configuration server again.

ICA9062 **Fatal configuration server error - main accept():** *system error message*

Explanation: Configuration server main routine failed, daemon ended.

User Response: Correct the specified system problem and start the configuration server again.

ICA9063 Configuration server error - *failing function*: **return code = 0x***function return code*

Explanation: The configuration server detected an error in the specified function. The daemon ends.

User Response: Correct the specified system problem and start the configuration server again.

ICA9064 Unknown option -*value* **ignored.**

Explanation: Specified option is not recognized.

ICA9065 Configuration server error - *failing function*: *system error message*

Explanation: The configuration server detected an error in the specified function. The daemon ends.

User Response: Correct the specified system problem and start the configuration server again.

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 being used.

System Action: The configuration server ends.

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 ends.

User Response: Correct the usage of the specified option and start the configuration server again.

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 ends.

User Response: See your Firewall administrator to verify the SSL environment.

Appendix B. Hardening and Backing Up Your Firewall Configuration

This appendix discusses hardening for the Windows NT system configuration and backing up your Firewall configuration.

Hardening for the Windows NT 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 the installation of the IBM Firewall software and edits the system resources that might compromise security.

Services that are not needed for the IBM Firewall configuration and that are a potential threat to security, are disabled. All non-TCP/IP protocols are disabled. These can be re-enabled using the Networking control panel.

The following services are disabled. The administrator can re-enable these services using the Services Control Manager in the control panel:

- Alerter Service
- ClipBook Server
- Computer Browser
- DHCP Client
- Directory Replicator
- FTP Publishing Service
- Gopher Publishing Service
- License Logging Service
- Messenger
- Network DDE
- Network DDE DSDM
- Remote Procedure Call (RPC) Locator
- Server (for example, LanMan Server)
- Spooler
- Telephony Service
- UPS
- World Wide Web Publishing Service

If the Firewall is a stand-alone machine, the following two services are also disabled:

- TCP/IP NetBios Helper
- Workstation (LanMan Workstation)

The following services are not disabled because the Microsoft DNS code depends on them:

- NT LM Security Support Provider
- Remote Procedure Call Service

We disable the right to do local logins (physically at the machine) for all users who are not in the Administrator group.

Backing Up Your Firewall Configuration

The Firewall stores all of its configuration files in `ROOTDIR\config`. If you want to backup your firewall configuration without backing up all of the Firewall files, back up the entire contents of the `ROOTDIR\config` directory.

If you want to restore a backed-up Firewall configuration, delete all of the existing files in the `ROOTDIR\config` directory and then restore the backed-up versions of the files. You will have to regenerate and activate the filter rules before the restored configuration will take effect.

Manual editing of these files is not supported.

Appendix C. 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.ietf.org/html.charters/ipsec-charter.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:RFCnnnn.TXT
RFC:RFCnnnn.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 RFCnnnn.TXT
or
    SEND RFCnnnn.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 D. IBM eNetwork Firewall Socks5.conf Configuration File Format

The configuration file **socks5.conf** is located in the IBM Firewall installation directory by default. If desired, you can edit this file using a text editor.

The **socks5.conf** configuration file is read the first time the server is invoked. (To refresh without stopping type `socks5.config`). This file contains all the information the IBM Firewall needs to determine which interface to use to reach a given address, whether to connect directly to a given address or to use another proxy server, and what requirements need to be met for a proxy connection to be made.

The following sections appear in the configuration file:

- Aliases
- Variables
- Modules
- Authentication
- Routing
- Proxies
- Access Control

In the Authentication, Routing, Proxies, and Access Control sections, lines are read in order until a match is made for that section: the order of the lines is very important. For a line to match, each entry within a line must match.

Specifying Ports

Ports can be specified using either a name, number, or range. Ranges begin with either a [or (and end with either a) or a] depending on whether or not the range is inclusive. Within the range delimiters should be two port specifiers (names or numbers), separated by a comma. The method of specifying ports is referred to as the *port pattern*.

Specifying Hosts

Host addresses and netmasks are often needed for specifying which hosts apply for a given rule. This method of specifying hosts is referred to as the *host pattern*. There are several ways to specify the host/mask pair:

Parameter	**Description**
hostIP/ mask	A host address ″ANDed″ with the mask must be the same as host IP ″ANDed″ with the mask. This is usually used to mask out the host portion of the address from the network or subnetwork portion.
-	Anything matches. All hosts are allowed.
n1	Equivalent to n1.0.0.0/255.0.0.0.
n1.n2	Equivalent to n1.n2.0.0/255.255.0.0.
n1.n2.n3	Equivalent to n1.n2.n3.0/255.255.255.0.
.domain.name	The host name must end with the string *.domain.name.*
a.host.name	The hostname must match exactly *a.host.name.*

There is also support for the older host pattern syntax, as described below. However, the newer method is recommended and easier to read.

Parameter	**Description**
hostIP/a	Anything matches (same as ″-″). All hosts are allowed.
hostIP/n	Network match. Masks out the host and subnet portions of the address, leaving only the network portion. The mask used to do this depends on the class of host IP address.
hostIP/s	Subnet match. Masks out the host portion of the address, leaving only the subnet and network portion. The mask used to do this depends on the class of host IP address.
hostIP/h	Host match. Equivalent to host IP.

Specifying Authentication Methods

The authentication methods shipped with the IBM Firewall are *ibmcram* and *ibmpwd.* Others can be added.

Authentication methods can be specified as a list of methods separated by commas. For a line to match, the chosen authentication method has to be reperesented by one of the methods in the list. This syntax is referred to as an *auth pattern*. The authentication method NULL is defined by default. Other methods may be included by loading the appropriate modules. A″-″ indicates any authentication method, including NULL, is acceptable.

Authentication Entries

The authentication entries indicate the types of authentication that can be used. The format is:

```
auth/ban   source-address  source-port
           auth-methods
```

Parameter	**Description**
auth/ban	Whether the authentication entries are authorized (auth) or not (ban).
source-address	A valid host pattern.
source-port	A valid port pattern.
auth-methods	A valid auth pattern.

The keyword ″ban″ indicates that authentication should not even be attempted with this host and has no valid use for the specified server.

If no auth/ban lines are specified, the default is that any authentication method is acceptable. If the permission for the connection is set to *deny* (the default), the connection would not be rejected until after authentication has been applied. In the SOCKS5 protocol, authentication takes place before authorization. You must decide based solely on the host, how that host is to authenticate.

Specifying Commands

Commands can also be specified as a list of commands separated by commas. This syntax is referred to as a command pattern. The commands defined are: connect, bind, udp, ping, and traceroute. Other commands may be added by loading the appropriate modules. A ″-″ (dash) indicates any command is acceptable.

Loading Modules

Modules allow custom expansion to server functionality by adding new authentication methods, commands, authorization checks, and content filters. The format is: *module stub filename options*

Parameter	Description
module	The identifier of the module to load.
stub	A module-dependent name prefix for accessing function names.
filename	The file name for the module to load.
options	Module-specific configuration information, if any.

Modules may define fields used elsewhere, so it is best to put module lines first. For example, authentication modules define authentication method names used in auth and permit lines.

Routing Entries

On machines with multiple network interfaces (hence, IP addresses), it is desirable to make sure that certain network interfaces are used in conjunction with certain addresses. This prevents ″IP spoofing″ (machines outside the network pretending to be machines inside the network), by making sure that inside machines use the inside network interface and outside machines use the outside network interface. It is also used by the SOCKS server in determining the network interface to bind on when accepting a BIND request, or when issuing a SENDTO request. If no entry matches, INADDR_ANY is used to bind, and a connection can be received on any interface. Single-homed hosts need not have routing entries: they are only necessary for machines with more than one network interface. The format is: **route** *dest-address dest-port interface-address*

Parameter	Description
route	Keyword to indicate the routing entries.
dest-address	A valid host pattern.
dest-port	A valid port pattern.
interface-address	Either the IP address of a network interface card, or the name of the network interface (for example, elnk31).

Variable Entries

The amount and types of logging and informational messages can be controlled by certain variables and flags in the configuration file. The format is: **set** *variable value*

Parameter	**Description**
set	Keyword to set the environment variable entries for local use.
variable	A valid environment variable. Refer to “Environment Variables” below for a listing of the available variables.
value	The value to assign.

Environment Variables

Environment Variable	**Description**
SOCKS5_BINDPORT [port]	Configures IBM Firewall to use the specified port, rather than the default of port 1080.
SOCKS5_RECVFROMANYONE	If UPD support is enabled, this allows the UPD clients to receive messages from unknown senders.
SOCKS5_USECLIENTSPORT	Configures IBM Firewall to proxy only if it can bind to the same port the client uses to send messages. This is necessary for proxying UDP connections when the server is streaming data to the client (sending messages to the client before the client sends messages to the server). An example of this usage would be RealAudio.
SOCKS5_MAXCHILD	The maximum number of concurrent threads.
SOCKS5_NOREVERSEMAP	Disables mapping of IP addresses to host names. If aliases are assigned in the configuration file, this would increase performance at the expense of logging information.
SOCKS5_NOSERVICENAME	Disables mapping of port numbers to service names. If aliases are assigned in the configuration file, this would increase performance at the expense of logging information.

Environment Variable	Description
SOCKS5_NOIDENT	Disables IDENT requests, even if compiled in. This is useful when you have a slow link to clients, and they are not using IDENTD. This will reduce the timeout periods.
SOCKS5_DEMAND_IDENT	Configures NULL authentication to fail if there is no IDENT response from clients. This is useful for ensuring that a user name is always associated with a connection request.

Proxy Entries

Proxy entries describe the addresses of SOCKS proxy servers. These lines tell the server how to contact a given host. If no lines match a host, the host is contacted directly. The format is: *proxy-type dest-addr dest-port proxy-addr proxy-port*

Parameter	Description
proxy_type	The type of proxy server. Valid entries are: • socks5 • socks4 • no proxy
dest-address	A valid host pattern.
dest-port	A valid port pattern.
proxy-address	Either the IP address or the name of the proxy server.
proxy-port	The proxy server port on which the SOCKS daemon is accepting connections.

Access Control Entries

The access control section determines whether a request to establish a connection is permitted or denied. There are two types of lines, permit lines and deny lines. Each entry on the line must match for the entire line to match. The format is:

```
permit auth cmd src-host dest-host src-port dest-port [userlist]
deny auth cmd src-host dest-host src-port dest-port [userlist]
```

Parameter	Description
auth	A list of authentication methods, specified by a valid auth pattern and auth entry.
cmd	A valid command pattern specifying the commands that are matched by this line.
scr-host	A valid host pattern for the source host.
dest-host	A valid host pattern for the destination host.
scr-port	A valid port pattern for the source host port.
dest-port	A valid port pattern for the destination host port.
userlist	A valid user pattern.

Filters

Filtering through a loaded module is performed by the filter directive. The format is:

```
filter  name auth cmd src-host dest-host src-port dest-port [userlist]
```

Parameter	Description
name	The identifier of the filter module.
auth	A list of authentication methods, specified by a valid auth pattern and auth entry.
cmd	A valid command pattern specifying the commands that are matched by this line.
scr-host	A valid host pattern for the source host.
dest-host	A valid host pattern for the destination host.
scr-port	A valid port pattern for the source host port.
dest-port	A valid port pattern for the destination host port.
userlist	A valid user pattern.

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*, GC31-8658
- *IBM eNetwork Firewall Reference*, SC31-8659
- *Guarding the Gates Using the IBM eNetwork Firewall for NT 3.2*, SG24-5209

Internet and World Wide Web Topics

- *A Comprehensive Guide to Virtual Private Networks, Volume 1: IBM Firewall, Server and Client Solutions*, SG24-5201
- *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

This industry publication pertains to Windows NT:

Cowart, Robert. *Windows NT Server 4.0 Administrator's Bible*. IDG Books Worldwide, 1996. (ISBN: 0764580094)

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

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. 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. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

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
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

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
Department TL3B/ Building 062
P.O. Box 12195
3039 Cornwallis
Research Triangle Park, NC 27709-2195
U.S.A.

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 information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

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
- DB2
- eNetwork
- IBM
- VisualAge

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States and/or other countries.

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States and/or other countries.

Other company, product, and service names may be trademarks or service marks of others.

Index

Readers' Comments — We'd Like to Hear from You

IBM eNetwork™ Firewall for Windows NT®
Reference
Version 3 Release 3

Publication No. SC31-8659-02

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	□	□	□	□	□

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	□	□	□	□	□
Complete	□	□	□	□	□
Easy to find	□	□	□	□	□
Easy to understand	□	□	□	□	□
Well organized	□	□	□	□	□
Applicable to your tasks	□	□	□	□	□

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

Company or Organization

Phone No.

Address

Readers' Comments — We'd Like to Hear from You

SC31-8659-02

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.

SC31-8659-02