IBM® SecureWay® Policy Director

Policy Director Administration Guide: Additions and Corrections

Version 3.0.1

IBM® SecureWay® Policy Director

Policy Director Administration Guide: Additions and Corrections

Version 3.0.1

Note

Before using this information and the product it supports, read the general information under “Appendix B. Notices” on page 45.

First Edition (January 2000)

This edition applies to version 3, release 0, modification 1 of the IBM® SecureWay® Policy Director product and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

About this document

This document contains updates to the material contained in the Policy Director Administration Guide, First Edition (October, 1999), Version 3, Release 0, Modification 0.

These updates include the following types of information:

- new topics to the administration guide
- enhancements to existing topics
- technical corrections to existing content

For convenient referencing, the additions and corrections contained in this document are organized according to the chapter structure of the original administration guide.

Chapter 1. Welcome to Policy Director

The following sections in Chapter 1 contain additions and corrections:

Introducing Policy Director

Policy Director core technologies

Authentication

Policy Director supports the following authentication mechanisms:

Secret Key:

- Kerberos V5
- LDAP

Public/Private Key

- Login over SSL via client-side X.509 certificate

The current document erroneously lists “login over an SSL browser using applications-specific username and password” as an authentication mechanism. This technique is actually not an authentication mechanism, but instead, a method of passing client identity information to WebSEAL. Once this identity information is received, WebSEAL provides actual authentication via Kerberos or LDAP.

Understanding the security model

Defining a security policy

Who can participate in the secure domain?

The graphic associated with this topic erroneously portrays a “security registry” database. Change the database label to “user registry”.

Applying security policy to a client request

The graphic associated with this topic erroneously portrays a “security registry” database. Change the database label to “user registry”.

Chapter 2. Authentication and credentials acquisition

The following sections in Chapter 2 contain additions and corrections:

Basic concepts of SSL authentication mechanism

Revised descriptions as follows:

Server authentication using server-side certificates

The following WebSEAL-specific description of the authentication process over SSL is very general. You can find full details of the SSL protocol in the appropriate RFCs.

1. A client requests a connection with a WebSEAL server.
2. In response, WebSEAL sends its public key via a signed server-side certificate. This certificate has been previously signed by a trusted third-party certificate authority (CA).
3. The client checks whether it can trust and accept the issuer of the certificate. The browser usually contains a list of root certificates from trusted certificate authorities. If the signature on the certificate from the WebSEAL server matches one of these root certificates, then the server can be trusted.
4. If there is no match for the signature, the browser informs its user that this certificate was issued by an unknown certificate authority. It is then the user's responsibility to accept or reject the certificate.
5. If the signature matches an entry in the browser's root certificate database, session keys are securely negotiated between the client and the WebSEAL server. The end result of this process is a secure channel over which the client can authenticate (for example, via username and password). After successful authentication, the client and server can continue to communicate securely over this channel.

Client authentication by using client-side certificates

The following WebSEAL-specific description of the authentication process over SSL is meant to be very general. You can find full details of the SSL protocol from other sources, such as the Netscape Web site.

1. A client requests a connection with a WebSEAL server.
2. In addition to the server authentication process, described in the previous section, the client sends its public key certificate to the WebSEAL server.
3. WebSEAL attempts to match the signature on the client certificate to a known certificate authority (CA). Like a client browser, the WebSEAL server maintains a list of root certificates from trusted certificate authorities.
4. If there is no match for the signature, WebSEAL will generate an SSL error code and send it to the client.
5. If there is a match for the signature, then the client can be trusted. WebSEAL passes the certificate on to a Credentials Acquisition Server (CAS) which completes the authentication process.
6. Session keys are securely negotiated between the client and the WebSEAL server. The end result of this process is a secure communication channel between the authenticated client and server.

Username and password authentication

This section erroneously portrays Basic Authentication and Forms-Based Login as authentication mechanisms. They are actually methods of providing client identity information to WebSEAL. WebSEAL then uses Kerberos or LDAP to provide true authentication of this client.

Credentials acquisition service overview

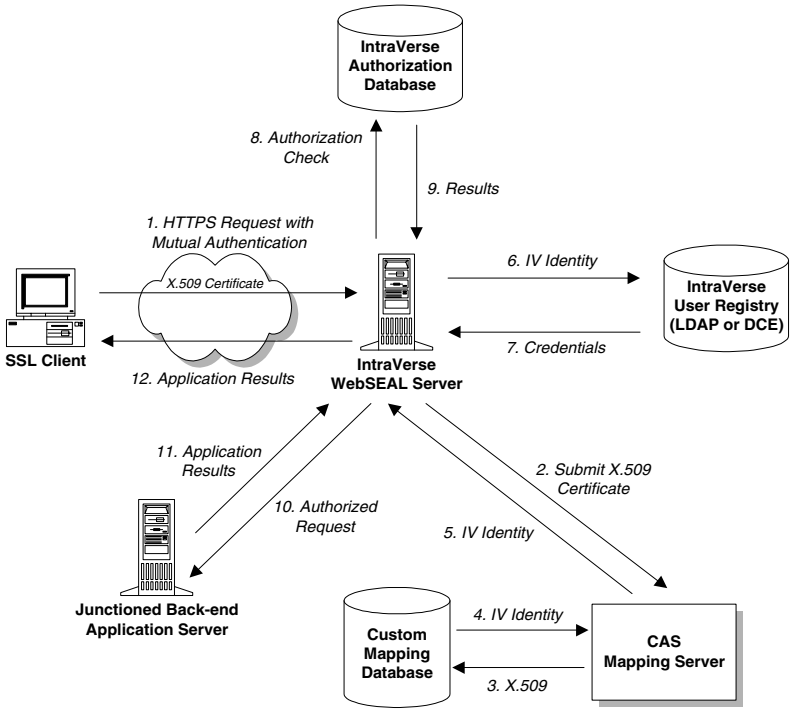
Introduction to the credentials acquisition service

The graphic associated with this topic erroneously portrays a “security registry” database. Change the database label to “user registry”.

X.509 certificate mapping mode

The graphic associated with this topic erroneously portrays a “security registry” database. Change the database label to “user registry”.

The final graphic is missing the numbering scheme that allows you to follow the sequence in the correct order. The correctly annotated version appears below:



Username mapping mode

The graphic associated with this topic erroneously portrays a “security registry” database. Change the database label to “user registry”.

Chapter 3. Understanding authorization

The following sections in Chapter 3 contain additions and corrections:

Network security policy

Network security-policy definition

Change references to the “security registry” to “user registry”.

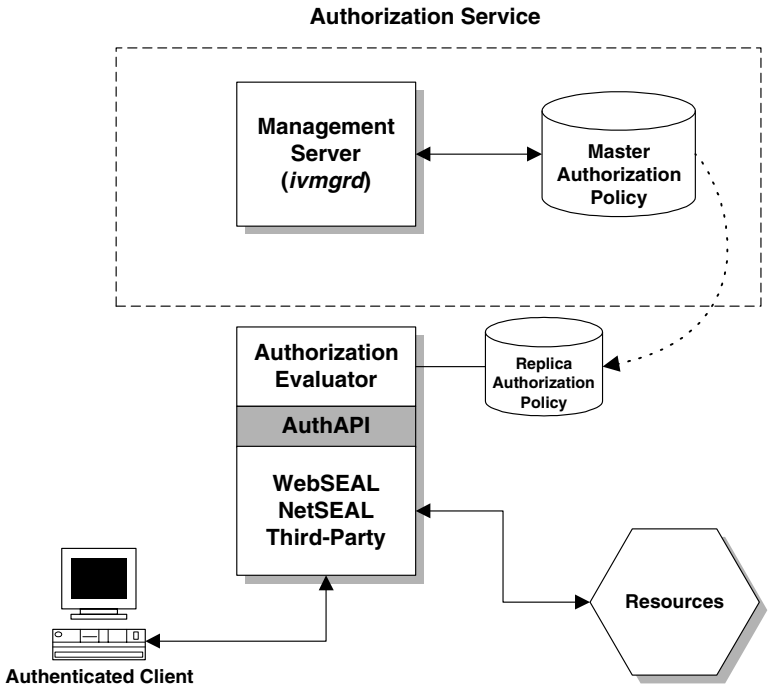
Policy administration

Change references to the “security registry” and “master registry” in the text and figure to “user registry”.

Policy Director Authorization API

Local cache mode

This section contains the wrong figure. The correct figure appears below:



Chapter 4. Introducing the Management Console

The following sections in Chapter 4 contain additions and corrections:

Management Console features

Pin view panel

This feature no longer exists.

Users management task

The User Detail view (not Groups view) provides three distinct object lists:

- Groups
- Resources
- Resource groups

ACLs management task

There is no List action button.

Management Console properties and controls

Add the following Console features:

Scroll Arrow Buttons for Tabs

Some Console views, such as the User Detail object list, contain multiple tabs that can display different lists of information. When the width dimension of the view window does not allow room for all available tabs, scroll arrow buttons appear on the lower right corner of the window. Click on the left and right arrow buttons to scroll the tabs.

In-Line Editing for Object List Items

The User Detail view contains single sign-on login and password fields for resource credentials. You activate these fields by clicking on the field area. Any existing data in the field becomes highlighted in gray. You can now edit the information in the field.

Specify and Invoke Queries

Some list views have query capabilities. Click on the binocular icon to open a query field. Enter the query conditions. You can use the asterisk (*) wildcard. Press Enter to display the results. The neighboring down-arrow icon performs a Get operation for all available items.

Chapter 6. Managing GSO resources, resource groups, and resource credentials

The following sections in Chapter 6 contain additions and corrections:

- Changing the GSO resource credential password

Changing the GSO resource credential password

Revised description of the **chpwd** utility:

"A user can change:

- the stored GSO password,
- change the user ID, or
- change both attributes

for a GSO resource or a GSO resource group using the Policy Director Web-based password tool, chpwd.exe."

Chapter 7. Understanding access control

The following sections in Chapter 7 contain additions and corrections:

ACL entry syntax

Permission sequence

This section erroneously lists the “p” (Proxy) permission. This permission no longer exists and has been replaced by the “f” (Forward) permission. This permission applies to NetSEAL scenarios only.

Regions of the namespace

NetSEAL namespace

NetSEAL permissions

The description of the “C” (Connect) permission should read as follows: “Connect through a NetSEAL server to a local or remote protected service.”

Chapter 8. Applying Access Control

The following sections in Chapter 8 contain additions and corrections:

- ACL management overview
- Object Space management overview

ACL management overview

Action buttons for ACL management tasks

The List action button does not exist.

Object Space management overview

Action buttons for Object Space management tasks

The List action button does not exist.

Chapter 9. Managing Proxy Users

The following sections in Chapter 9 contain additions and corrections:

- Using the ivadmin policy commands for proxy user management

Using the ivadmin policy commands for proxy user management

Managing password policies

- Delete all reference to the **policy set|get password-expiry-date** command.
- The **policy set|get password-expiry-warn** command has been renamed to **policy set|get number-warn-days**

Chapter 10. Managing the Policy Director servers

The following sections in Chapter 10 contain additions and corrections:

- UNIX: Stopping and starting Policy Director servers
- Windows: Stopping and starting Policy Director servers

UNIX: Stopping and starting Policy Director servers

Stopping using the iv script

The use of the kill command for manual shutdown is not recommended.

The shutdown order for Policy Director processes is not correct. The correct order is:

1. ivacld
2. cdas
3. secmgrd
4. ivmgrd
5. dsb

Starting using the iv script

The startup order for Policy Director processes is not correct. The correct order is:

1. dsb
2. ivmgrd
3. secmgrd
4. cdas
5. ivacld

Windows: Stopping and starting Policy Director servers

The Policy Director X.509 Authorization Server process illustrated in Step 2 does not exist. The Cross Domain Authentication Service process appears instead.

Chapter 11. Managing the Authorization service

The following sections in Chapter 11 contain additions and corrections:

- Managing permissions
- Managing Local Cache Mode API Applications
- Defining external authorization services

Managing permissions

This section heading creates an inappropriate new section. The information contained in this inappropriate section is actually a continuation of the section: “Defining custom ACL permissions”

Managing Local Cache Mode API Applications

The following material represents an addition to this chapter:

The Policy Director Authorization Application Programming Interface (API) allows Policy Director applications and third-party applications to query the Authorization Service to make authorization decisions. Any application that uses the Authorization API must have access to a replica of the master authorization policy database. The information contained in this database is essential for the authorization evaluation process.

Third-party applications can operate in remote cache mode, where the application relies on the database replica maintained by the Authorization Server (ivacld). An application can also run in local cache mode, where it maintains its own replica of the authorization policy database.

You must manually register any application using the Authorization API in local cache mode with the Policy Director Authorization Service. The Management Server, a component of the Authorization Service, must know the location of any local cache mode Authorization API application so it can update the replica authorization policy database associated with it.

Two general steps are required to set up an application to use the Authorization API in local cache mode with its own replica policy database:

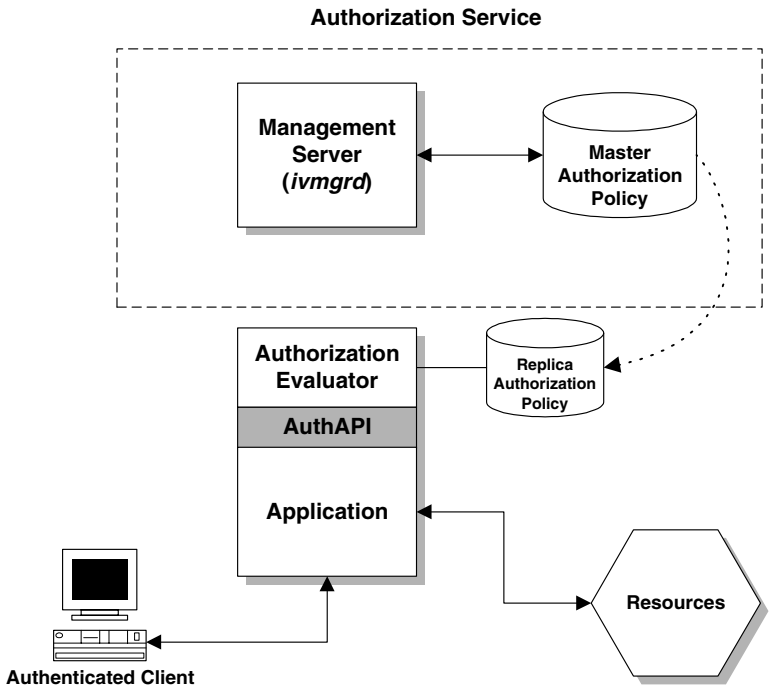
1. Write the application program that uses the Authorization API.

   Refer to the *Policy Director 3.0 Programming Guide and Reference*.

2. Register the application with the Policy Director Authorization Service.

   Refer to: “Registering Local Cache Mode API Applications” in this chapter.

For further background information, see “The Policy Director Authorization API” in Chapter 3.



Registering Local Cache Mode API Applications

Use the **ivadmin server register dbreplica** command to inform the Authorization Service (specifically, the Management Server) of the existence and location of applications using the Authorization API in local cache mode. The following syntax applies:

```
ivadmin> server register dbreplica <server-name> <ns-location> <server-
principal> <server-host>
```

where:

server-name	A name (or label) for this application. This is the name that appears in the display of the object space on the Management Console and in the **ivadmin server list** command.
ns-location	The RPC entry in the CDS namespace where the application exports its RPC bindings.
server-principal	The name of the DCE principal representing this application process.
server-host	The DNS name or IP address of the machine where this application process resides.

For example:

```
ivadmin> server register dbreplica print-spooler /.:/subsys/spooler
spooler clipper
```

registers an application named print-spooler with the Authorization Service.

The RPC entry in the CDS namespace where print-spooler exports its RPC bindings is /.:/subsys/spooler.

The DCE principal name for the application is spooler.

The DNS name of the machine where this application process resides is clipper.

Administration Tasks for Authorization API Applications

You can use the **ivadmin server** commands to list information about any registered Authorization API application.

List Registered Applications and Servers

The following command lists all registered applications and servers in the secure domain:

```
ivadmin> server list
```

Display Application Status

The following command indicates whether the specified application is running or stopped, and if the replica authorization policy database for that application has been updated with the latest changes:

```
ivadmin> server status <server-name>
```

Show Application Details

The following command displays the specified properties for the application, such as name, description, hostname, NS location, principal name, and root Web address:

```
ivadmin> server show <server-name>
```

Delete a Registered Application

The following command deletes a registered Authorization API application:

```
ivadmin> server delete <server-name>
```

Defining external authorization services

Registering an external authorization service

Replace **server-principal** with **server-group**. The **server-group** parameter is the name of the DCE group whose membership includes the external authorization server process.

Chapter 12. Logging and auditing server activity

The following sections in Chapter 12 contain additions and corrections:

Standard HTTP logging

Configuring standard HTTP logging

Specifying the maximum log file size

Replace this section with the following new section:

Specifying Log File Rollover Thresholds

The **logsize** parameter specifies the maximum size to which each of the HTTP log files may grow and has the following default value:

```
[wand]
logsize = 2000000
```

Note that this parameter also impacts the Policy Director wand_audit_log audit trail file.

When a log file reaches the specified value — known as its *rollover threshold* — the existing file is backed up to a file of the same name with an appended current date and timestamp. A new log file is then started.

The various possible **logsize** values are interpreted as follows:

- If the **logsize** value is less than zero (< 0), then a new log file is created with each invocation of the logging process and every 24 hours from that instance.
- If the **logsize** value is equal to zero (= 0), then no rollovers are performed and the log file grows indefinitely. If a log file already exists, new data is appended to it.
- If the **logsize** value is greater than zero (> 0), then a rollover is performed when a log file reaches the configured threshold value. If a log file already exists at startup, new data is appended to it.

Specifying the Frequency for Flushing Log File Buffers

New section:

Log files are written to buffered data streams. If you are monitoring the log files in real time, you may want to alter the frequency with which the server forces a flush of the log file buffers.

By default, log files are flushed every 20 seconds:

```
[wand]
logflush = 20
```

If you specify a negative value, a flush will be forced after every record is written.

Policy Director authorization audit trail files

Example Management Server audit trail file

New output format:

```
START RECORD
1999-08-10-08:32:30.161I-----
  Protected object: /Management/ACL
  Requested permissions: 0x00000400
  User:
  Principals:
IV_DCE_V3.0 00000064-2380-21d3-8900-0a80007daa77
  qop: unknown (0)
  result: authorized
END RECORD

START RECORD
1999-08-24-04:41:40.580I----
  Protected object: /WebSEAL
  Requested permissions: 0x00000100
  User: application_administrator
  Principals:
IV_LDAP_V3.0 00000064-35ee-21d2-a000-0800207b48c55
  qop: none
  result: authorized
END RECORD

START RECORD
1999-08-25-05:47:20.679I----
  Protected object: /WebSEAL/sun
  Requested permissions: 0x00000100
  User: application_administrator
  Principals:
IV_DCE_V3.0 00000064-35ee-21d2-a000-0800207b48c55
  qop: none
  result: authorize
END RECORD

START RECORD
1999-08-26-11:46:45.340I----
  Protected object: /WebSEAL/sun/icons
  Requested permissions: 0x00000100
  User: application_administrator
  Principals: IV_UNAUTH_V3.0
  qop: none
  result: authorize
```

```
END RECORD

START RECORD
1999-08-27-03:22:28.682I----
  Protected object: /WebSEAL
  Requested permissions: 0x00000100
  User: application_administrator
  Principals:
IV_LDAP_V3.0 00000064-35ee-21d2-a000-0800207b48c55
  qop: none
  result: authorize
END RECORD
```

WebSEAL audit trail file

WebSEAL auditing

Specifying the maximum log file size

Replace this section with the following two new sections:

Specifying Log File Rollover Thresholds

Refer to the equivalent discussion in “Configuring Standard HTTP Logging”.

Specifying the Frequency for Flushing Log File Buffers

Refer to the equivalent discussion in “Configuring Standard HTTP Logging”.

Policy Director management command audit trail file

Replace this entire section with the following information:

Each Policy Director server can capture audit events whenever any management-related auditable activity occurs. Audit events are saved as audit records that document the specific activity of that server. An audit trail file is made up of multiple audit records.

The following table illustrates the relationship between the Management Server and its associated audit trail file:

Server	Process	Management Audit File
Management Server	**ivmgrd**	Defined in ivmgrd.conf: mgrlog= <*install-path*>/ivmgrd/log/mgraudit.log

The responsibilities of the Management Server include maintaining the master authorization policy database. This database includes the description of the protected object namespace for the secure domain, ACL policy templates, and where ACLs are attached to objects.

Any management command event, from the Management Console or the **ivadmin** utility, can be captured in the mgraudit.log file.

Configuring Policy Director Server Management Auditing

Parameters for configuring Policy Director server management audit trail files are located in the [ivmgrd] stanza of the ivmgrd.conf configuration file.

Enabling and Disabling Management Auditing

By default, Policy Director server management auditing is enabled:

```
[ivmgrd]
logmgr = yes
```

To disable auditing, set:

```
logmgr = no
```

Note: There must be no space after the "yes" or "no" when editing this parameter in the ivmgrd.conf file.

Specifying the Log File Location

The default location of the Policy Director server management auditing file is:

UNIX:

```
[ivmgrd]
mgrlog = /opt/intraverse/ivmgrd/log/mgraudit.log
```

Windows:

```
[ivmgrd]
mgrlog=C:\Program Files\DASCOM Inc\IntraVerse\ivmgrd\log\mgraudit.log
```

Specifying Log File Rollover Thresholds

The **logsize** parameter specifies the maximum size to which the Policy Director server management audit file may grow and has the following default value:

```
[ivmgrd]
logsize = 2000000
```

Refer to the equivalent discussion in "Configuring Standard HTTP Logging"

Specifying the Frequency for Flushing Log File Buffers

By default, log files are flushed every 20 seconds:

```
[ivmgrd]
logflush = 20
```

Refer to the equivalent discussion in "Configuring Standard HTTP Logging"

Audit Record Contents

The audit records will be written in records tagged with XML style bracketing. An audit event captures the following information:

- **Originator ID**

  Derived from the user credentials information described in the *Policy Director 3.0 Programming Guide and Reference*. The field is delimited by the <C> tag and may contain the following sub fields as appropriate:

  User Information: Delimited by the <U> tag. A free format description string.

  Authorized Principal ID: Delimited by the <P> tag. This field value is made up of a string indicating the authentication method used – either IV_UNAUTH_V3.0, IV_LDAP_V3.0, or IV_DCE_V3.0

  In the case of LDAP or DCE authentication, the principal's UUID will also be shown following the authentication string. For example:

  ```
  <C><U>application_administrator</U><P>IV_LDAP_V3.0 00000064-35ee-
  21d2-a000-0800207b48c5</P></C>
  ```

- **Event ID**

  A number which uniquely identifies a management command.

  Only a subset of server management activity are audited. Commands which do not result in an update, such as **show** and **list**, are not logged.

  Valid values include:

ACL Management Commands	
ACL_SET	3002
ACL_DELETE	3003
ACTION_SET	3007
ACTION_DELETE	3008
Object Management Commands	
OBJ_ACL_SET	3101
Server Management Commands	
SERVER_SET	3200
SERVER_DELETE	3205
SERVER_LIST	3206
SERVER_ENABLE	3207
SERVER_DISABLE	3208
NetSEAL Management Commands	
NS_JCT_ADD	3300

NS_JCT_DELETE	3301
NS_PORT_ADD	3303
NS_PORT_DELETE	3304
NS_NET_ADD	3306
NS_NET_DELETE	3307
NS_PORTALIAS_ADD	3309
NS_PORTALIAS_DELETE	3310
Administration, User, and Group Management Commands	
USER_CREATE	3401
USER_MODDESC	3403
USER_MODPWD	3404
USER_MODAUTHMECH	3405
USER_MODACCVALID	3406
USER_MODPWDVALID	3407
USER_DELETE	3408
GROUP_CREATE	3414
GROUP_IMPORT	3415
GROUP_MODDESC	3416
GROUP_MODADD	3417
GROUP_MODREMOVE	3418
GROUP_DELETE	3419
USER_MODGSOUSER	3425
USER_SET	3426
GROUP_SET	3427

- **Command outcome**

  A number corresponding to the status code returned to the caller.

  Tag <O>

- **Time stamp**

  A record of the time when the command was completed – in Universal Time Format.

  Tag <D>

- **Command argument vector**

  A representation of the command input arguments.

  Tags <V> and <A>

Example Audit Trail File for Management Server

```
<E><D>1999-09-20-23:31:08.161I-----
</D><I>3003</I><O>0</O><C><P>IV_DCE_V3.0 00000064-2380-21d3-8900-
0a80007daa77</P></C><V><A><B>1default-
replica</B><B>0</B><B></B></A></V></E>

<E><D>1999-09-20-23:31:09.142I-----
</D><I>3003</I><O>0</O><C><P>IV_DCE_V3.0 00000064-2380-21d3-8900-
0a80007daa77</P></C><V><A><B>1default-
root</B><B>0</B><B></B></A></V></E>

<E><D>1999-09-20-23:31:09.824I-----
</D><I>3003</I><O>0</O><C><P>IV_DCE_V3.0 00000064-2380-21d3-8900-
0a80007daa77</P></C><V><A><B>1default-
management</B><B>0</B><B></B></A></V></E>
```

DCE server audit trail files

sec_audit trail example

Missing **dcecp** command line:

```
dcecp> audtrail show /opt/dcelocal/var/security/sec_audit_trail
```

Chapter 13. WebSEAL: Setting up authentication

The following sections in Chapter 13 contain additions and corrections:

Ensuring secure communication over SSL

The digital ID of the server in the diagram should be labeled “Verisign”, NOT “IBM”. This is the public server certificate issued to www.ibm.com by the Verisign Certificate Authority.

Policy Director Credentials Acquisition Service

Introducing the Policy Director CAS

An external user registry is missing from the diagram.

Chapter 14. WebSEAL: General administration tasks

The following sections in Chapter 14 contain additions and corrections:

- Enabling and disabling WebSEAL security
- Managing the Web space
- Configuring HTTP error messages

Enabling and disabling WebSEAL security

The **hostname** variable is missing from the **ivadmin** command examples:

```
ivadmin> server enable /WebSEAL/<hostname>
```

```
ivadmin> server disable /WebSEAL/<hostname>
```

Managing the Web space

Add the following new section:

Configuring Web document caching

Clients can often experience extended network access time and file downloading time due to poor Web document retrieval performance. Poor performance can occur because the WebSEAL server is waiting for documents retrieved from junctioned back-end servers or even slow local storage.

The Web document caching feature allows you to configure commonly accessed Web document types to be stored in the memory of the WebSEAL server. Clients will experience much faster response to follow-up requests for documents that have been cached in the WebSEAL server.

When you configure WebSEAL for Web document caching, you identify the following three parameters:

- Document MIME type
- Type of storage medium
- Size of storage medium

You define Web document caching in the [wand-caches] stanza of the iv.conf configuration file. The following syntax applies:

UNIX: `<mime-type> = <cache-type>:<cache-size>`

Parameter	Description
mime-type	Represents any valid MIME type conveyed in an HTTP "Content-Type:" response header. This value may contain a wildcard (*). A value of */* represents a default object cache that will hold any object that does not correspond to an explicitly configured cache.
cache-type	Specifies the type of storage medium to use for the cache. This release of Policy Director supports only "memory" caches.
cache-size	Specifies the maximum size (in kilobytes) to which the given cache can grow before objects are removed according to a "Least Recently Used" algorithm.

Examples:

```
text/html = memory:2000

image/* = memory:5000

*/* = memory:1000
```

The Web document caching mechanism observes these conditions:

- Caching only occurs when a cache is defined.
- No caches are defined at installation.
- If you do not specify a default cache, documents which do not match any explicit cache are not cached.

Flush All Caches

You can use the **wandmgr** command line utility to flush all configured caches. The utility does not allow you to flush individual caches.

You must login to the secure domain using **dce_login** before you can use **wandmgr**.

To flush all Web document caches, enter the following command:

UNIX: `# wandmgr <server-name> cache flush all`

Windows: `MSDOS> wandmgr <server-name> cache flush all`

Configuring HTTP error messages

Add the following new error message:

Filename	Title	Description	
132120c8.html	Authentication Failed	Credentials could not be retrieved for the client certificate used. Possible reasons include: the user supplied an incorrect certificate the certificate has been revoked the user's credentials are missing from the authentication database	

Chapter 15. WebSEAL: smart junction administration

The following sections in Chapter 15 contain additions and corrections:

Using junctioncp to manage smart junctions

Creating a new junction for an initial server

- **junctioncp** now supports **tcpproxy** and **sslproxy** types (**-t** option)
- junctioncp no longer supports the **-2** option (forcing SSL version 2 communication)

The following options are missing from the current documentation:

<options>		
	TCP and SSL junction options (use with -t tcp or ssl)	
	-u <*UUID*>	Specifies the UUID of a back-end server connected to WebSEAL via a stateful junction (-s). See “Specifying Back-End Server UUIDs for Stateful Junctions”.
	Local junction options (use with -t local)	
	-d <*dir*>	Local directory to junction. Required.
	Proxy junction options (required with -t tcpproxy and -t sslproxy)	
	-H <*hostname*>	The DNS hostname or IP address of the proxy server.
	-P <*port*>	The TCP port of the proxy server.

Adding an additional server to an existing junction

The following options are missing from the current documentation:

<options>		
	TCP and SSL junction options (use with -t tcp or ssl)	
	-u <*UUID*>	Specifies the UUID of a back-end server connected to WebSEAL via a stateful junction (-s). See “Specifying Back-End Server UUIDs for Stateful Junctions”.

	Proxy junction options (required with -t tcpproxy and -t sslproxy)	
	-H <*hostname*>	The DNS hostname or IP address of the proxy server.
	-P <*port*>	The TCP port of the proxy server.

Creating TCP and SSL Proxy Junctions

This section is an addition to the current documentation:

WebSEAL smart junction technology includes the ability to traverse network topologies that use HTTP or HTTPS proxy servers. You can configure the junction to handle requests as standard TCP communication or protected SSL communication.

The **junctioncp create** command requires one of the following arguments to the **type** option to establish either a TCP-based or SSL-based junction through a proxy server:

- **tcpproxy**
- **sslproxy**

Both **junctioncp create** and **junctioncp add** commands require the following options to identity the proxy server and the target Web server:

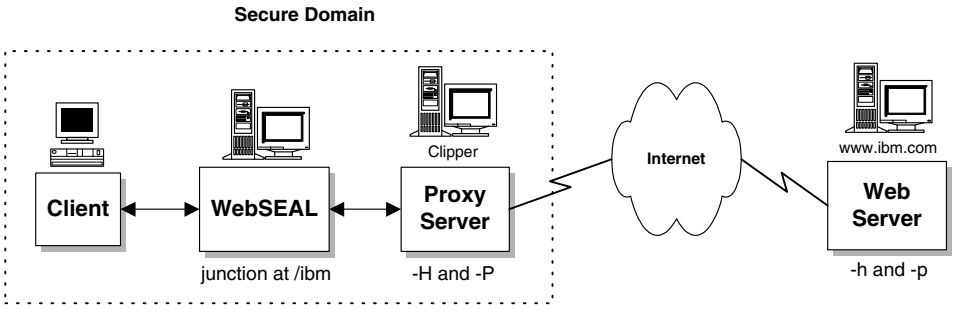
-H <*hostname*>	The DNS hostname or IP address of the proxy server.
-P <*port*>	The TCP port of the proxy server.
-h <*hostname*>	The DNS hostname or IP address of the target Web server.
-p <*port*>	The TCP port of target Web server. Default is 80 for TCP junctions; 443 for SSL junctions.

Example TCP proxy junction (entered as one line):

```
junctioncp> create -t tcpproxy -H clipper -P 8081 -h www.ibm.com -p 80
/ibm
```

Example SSL proxy junction (entered as one line):

```
junctioncp> create -t sslproxy -H clipper -P 8081 -h www.ibm.com -p 443
/ibm
```



Stateful Junction Support

This section replaces "Maintaining a state (**-s** option):

Most Web-enabled applications maintain a "state" for a sequence of HTTP client requests. This state is used, for example, to:

- Track a user's progress through the fields in a data entry form generated by a CGI program
- Maintain a user's context when performing a series of database inquiries
- Maintain a list of items in an online shopping cart application where a user randomly browses and selects items to purchase

Servers that run Web-enabled applications can be replicated in order to improve performance through load sharing. When the Policy Director server provides a smart junction to these replicated back-end servers, it must ensure that all the requests contained within a client session are forwarded to the correct server, and not distributed among the replicated back-end servers according to the load balancing rules.

By default, Policy Director balances back-end server load by distributing requests across all available replicated servers. Policy Director uses a "least-busy" algorithm.

The **junctioncp create** command with the **-s** flag overrides this load balancing rule and creates a "stateful junction" that ensures a client's requests are forwarded to the same server throughout an entire session. When the initial client request occurs, WebSEAL places a cookie on the client system that contains the UUID of the designated back-end server. When the client makes future requests to the same resource, the cookie's UUID information ensures that the requests are consistently routed to the same back-end server.

The **-s** option is appropriate for a single front-end WebSEAL server with multiple junctioned back-end servers. Note that once the first junction is created as stateful, **junctioncp add** is used without the **-s** option to junction the remaining replicated back-end servers.

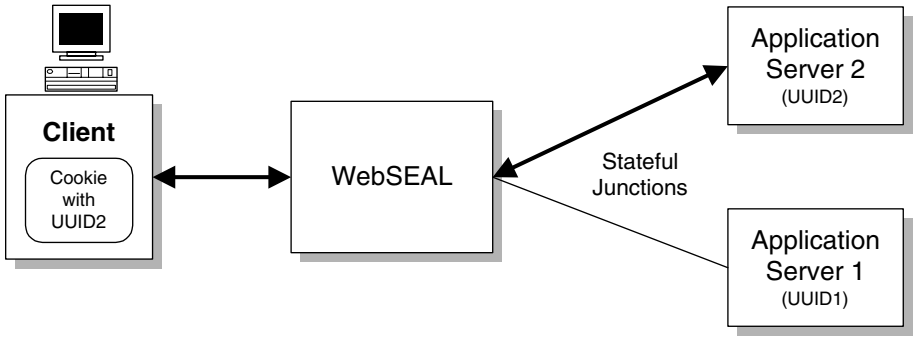
If the scenario involves multiple front-end WebSEAL servers, all junctioned to the same back-end servers, you must use the **-u** option to correctly specify each back-end server UUID to each front-end WebSEAL server. See Section "Specifying Back-End Server UUIDs for Stateful Junctions".

Specifying Back-End Server UUIDs for Stateful Junctions

This section is an addition to the current documentation:

When a new junction is created to a back-end application server, WebSEAL normally generates a Unique Universal Identifier (UUID) to identity that back-end server. This UUID is used internally and also to maintain stateful junctions (**junctioncp create -s**).

When the initial client request occurs, WebSEAL places a cookie on the client system that contains the UUID of the designated back-end server. When the client makes future requests to the same resource, the UUID information for the cookie ensures that the requests are consistently routed to the same back-end server.



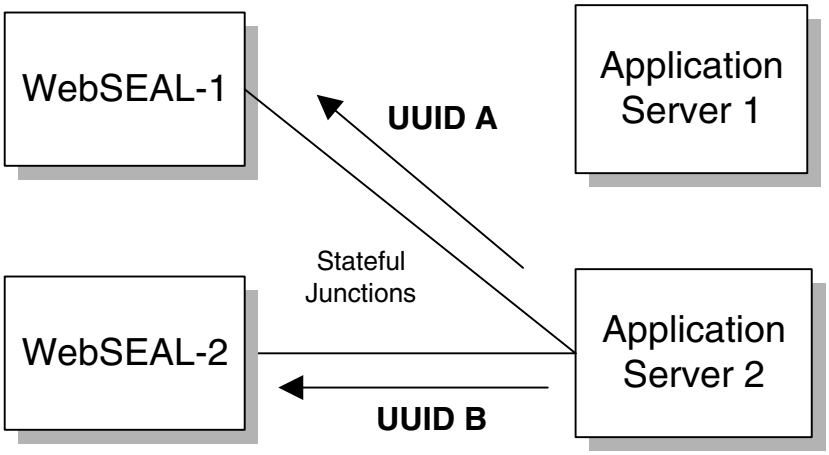
The handling of stateful junctions becomes more complex when there are multiple front-end WebSEAL servers junctioned to multiple back-end servers. Normally, each junction between a front-end WebSEAL server and a back-end server generates a unique UUID for the back-end server. This means a single back-end server will have a different UUID on each front-end WebSEAL server.

Multiple front-end servers require a load balancing mechanism to distribute the load between the two servers. For example, an initial state could be established to a back-end server through WebSEAL server 1 using a specific UUID.

However, if a future request from the same client is routed through WebSEAL server 2 by the load balancing mechanism, the state will no longer exist, unless WebSEAL server 2 uses the same UUID to identity the back-end server. Normally, this will not be the case.

The **junctioncp create -u** option allows you to supply the same UUID for a specific back-end server to each front-end WebSEAL server.

As an example, consider two replicated front-end WebSEAL servers, each with a stateful junction to two back-end application servers. When you create the stateful junction between WebSEAL server 1 and back-end server 2, a unique UUID (UUID A) is generated to identity back-end server 2. However, when a stateful junction is created between WebSEAL server 2 and back-end server 2, a new and different UUID (UUID B) is generated to identify back-end server 2.



A state established between a client and back-end server 2, via WebSEAL server 1 will fail if a subsequent request from the client is routed through WebSEAL server 2.

Apply the following process for specifying a UUID during the creation of a junction:

1. Create a junction from WebSEAL server 1 to each back-end server.

   Use **junctioncp create -s** and **junctioncp add**.

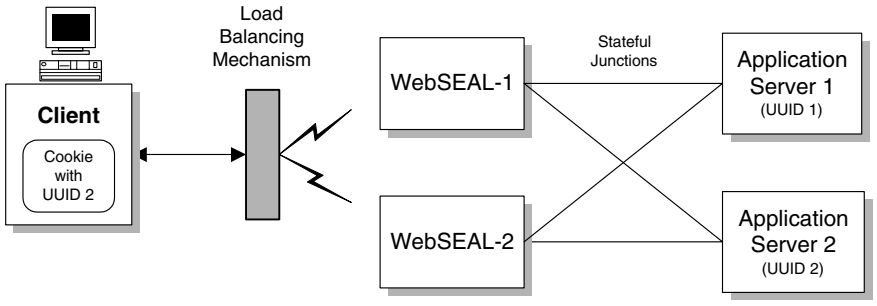
2. List the UUID generated for each back-end server during Step 1.

   Use **junctioncp show**.

3. Create a junction from WebSEAL server 2 to each back-end server and specify the UUIDs identified in Step 2.

   Use **junctioncp create -s -u** and **junctioncp add -u**.

In the following figure, back-end server 1 is known by both WebSEAL-1 and WebSEAL-2 as UUID 1. Back-end server 2 is known by both WebSEAL-1 and WebSEAL-2 as UUID 2.



Example:

In the following example, back-end server 1 is called svr1; back-end server 2 is called svr2.

UNIX: `# junctioncp -e WebSEAL-1`
Windows: `MSDOS> junctioncp -e WebSEAL-1`

```
junctioncp> create -t tcp -h svr1 -s /mnt
junctioncp> add -h svr2 /mnt
junctioncp> show
```

(This reveals UUID1 and UUID2)

UNIX: `# junctioncp -e WebSEAL-2`
Windows: `MSDOS> junctioncp -e WebSEAL-2`

```
junctioncp> create -t tcp -h svr1 -u <UUID1> -s /mnt
junctioncp> add -h svr2 -u <UUID2> /mnt
```

When a client establishes a stateful connection with back-end server 2, it receives a cookie containing UUID2. The above example now ensures that the client will always connect to back-end server 2, regardless of whether future requests are routed through WebSEAL-1 or WebSEAL-2.

Creating secure SSL smart junctions

Configuring a secure SSL junction

The -2 option is no longer valid.

Reviewing examples of SSL junctions

The -2 option is no longer valid.

Integrating GSO and WebSEAL single sign-on

Obtaining authentication information from GSO

Revised information:

-T <resource/ resource-group>	Specifies the GSO resource or resource group. The resource name used as the argument to this option must exactly match the resource name as listed in the GSO database. Required for gso junctions.

Using smart junctions

Controlling CGI processing (x permission)

Revised section header and content as follows:

Exceptions to Enforcing Permissions Across Junctions

Certain Policy Director permissions are not enforceable across a junction. You cannot control, for example, the execution of a CGI script with the **x** permission, or a directory listing with the **l** permission. WebSEAL has no means of accurately determining whether or not a requested object on a back-end server is, for example, a CGI program file, a dynamic directory listing, or a regular HTTP object.

Access to objects across junctions, including CGI programs and directory listings, is only controlled through the **r** permission.

Chapter 16. WebSEAL: Application Integration

The following sections in Chapter 16 contain additions and corrections:

Supporting CGI programming

New section as follows:

Windows: Supporting WIN32 Environment Variables

Windows does not automatically make all of its system environment variables available to processes such as CGI applications. Typically, the system environment variables you require will be present.

However, if any Windows system environment variables that you require are not present in the CGI environment, you can explicitly make them available to CGI programs through the iv.conf configuration file. (Note that the Policy Director environment variables mentioned above are automatically available on all platforms).

Add any of the required Windows system environment variables to the [inherited-env] stanza of the iv.conf file. Use the following format:

```
ENV = <variable-name>
```

Note that two examples are provided in the stanza (but commented out).

```
[inherited-env]
#ENV = SystemRoot
#ENV = SystemDrive
```

By uncommenting these example lines, the **SystemRoot** and **SystemDrive** variables will be present in a CGI environment.

Chapter 17. NetSEAL: Overview

The following sections in Chapter 17 contain additions and corrections:

Introducing NetSEAL

NetSEAL network segments

Reworded bullet items:

NetSEAL cannot provide protection to a network port unless that port is defined (created) through the **ivadmin netseal port add** command.

NetSEAL follows the same connection decision process for both local and remote applications:

- If the port is defined, does the user have the proper permission for access?
- If the port is not defined, and the communication originates over a GSS tunnel, reject the connection.
- If the port is not defined, and the communication originates from an outgoing trap, allow the connection.

NetSEAL also separates the issues of incoming information required by the Security Manager and outgoing connection processing. In other words, the Security Manager does not need to know whether the connection request has been trapped locally or trapped by a remote NetSEAT client.

Illustrating client-to-NetSEAL services

Incoming tunneled connection to protected host

Revised transaction description:

1. Can the user connect to the requested defined port on the destination server (based on its permissions)?

   No — Reject the connection request.

   Yes — Continue.

2. Is the destination a Policy Director server?

   Yes — Establish a secure tunnel to the server. Establish a TCP connection to the requested port.

   No — Establish a TCP connection to the requested port.

Incoming TCP connection to Policy Director server

Revised description:

This scenario considers the situation for a non-NetSEAT TCP client user. Such a client is recognized by Policy Director as unauthenticated. If the requested port is not defined (**ivadmin netseal port add**), access to the port is allowed. If the port is defined, the Security Manager checks the ACL protecting that port for unauthenticated access.

This configuration protects direct access to network services. An external authorization service could use the client's IP address to determine access rights.

The NetSEAL server completes the transaction in the following manner:

1. Is the request trapped by Policy Director (a defined port with permissions)?

   No — Allow incoming connection.

   Yes — Pass request to Security Manager (secmgrd).

2. Are unauthenticated requests on the port permitted?

   Yes — Establish a TCP connection to the requested port.

   No — Reject the connection request.

Illustrating NetSEAL-to-NetSEAL services

Outgoing connection to protected host

This section header is incorrect. The title should read: "Outgoing connection to an application server".

Revised description as follows:

A connection between a NetSEAL server and a third-party server is made over TCP and is therefore not a secure connection. Because there is no NetSEAL server on the back-end third-party server, Policy Director can only permit or deny a connection to the back-end server; Policy Director cannot secure the communication across the connection.

The Policy Director server completes the transaction in the following manner:

1. Is the requested port on the destination machine defined and protected with permissions?

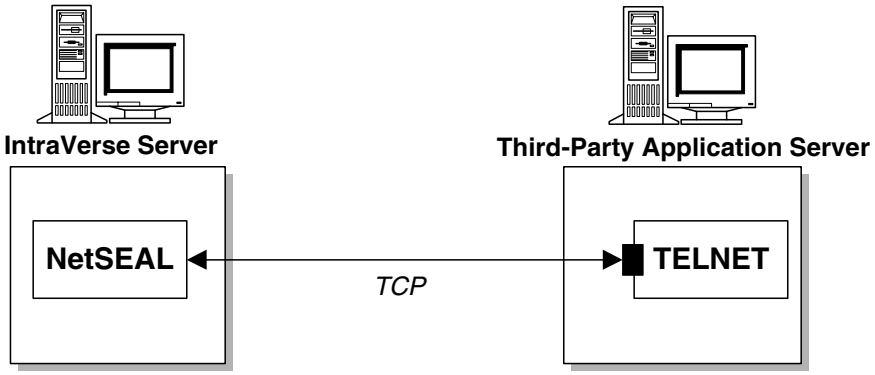
   No — Allow outgoing connection.

   Yes — Pass request to Security Manager (secmgrd).

2. Does the user have permission to access the requested port on the destination server?

   No — Reject the connection request.

   Yes — Establish a TCP connection to the requested port.


IntraVerse Server
Third-Party Application Server
NetSEAL
TELNET
TCP

Chapter 18. NetSEAL: General administration tasks

The following sections in Chapter 18 contain additions and corrections:

Managing protected ports

The example command line should read:

```
ivadmin> netseal port add west 23
```

Sample NetSEAL configurations

This new section follows the section: “Managing protected port aliases”

Listed below are several sample NetSEAL configurations that control access to TCP/IP applications. All examples are based on the TELNET application. The name of the Policy Director server is truffle.

The system administrator can define an ACL template to permit NetSEAT users access to NetSEAL services. The example configurations use the following ACL template:

```
ivadmin> acl create auth-access
ivadmin> acl modify auth-access description “Authenticated Access”
ivadmin> acl modify auth-access set user cell_admin abcTdmvC
ivadmin> acl modify auth-access set unauthenticated “”
ivadmin> acl modify auth-access set any-other “TC”
```

The examples use the following port-alias:

```
ivadmin> netseal port-alias add 23 telnet
```

Note: If the request in these examples is from a NetSEAT client, the user is authenticated. However, if the request is from a NetSEAL server, the user will be unauthenticated.

Note: In the examples below the ACL has been applied to the individual ports. This is not always necessary. ACLs higher in the hierarchy will be inherited.

Note: The configuration database is a distributed database. There may occasionally be delays between issuing an ivadmin command and the command taking effect. The server status command will show the status of the database.

Note: On AIX systems **inetd** may already be listening on a port. In this case it will be necessary to stop and then restart **inetd** after adding the port.

Controlling Access to Policy Director Servers

To control access to a Policy Director server from a NetSEAT client or a NetSEAL server, you must create a port entry for the Policy Director server.

For these examples, the Port Range in the Host Security will be the TELNET port (port 23).

```
ivadmin> netseal port add truffle telnet

ivadmin> acl attach /NetSEAL/truffle/telnet auth-access
```

This will allow only authenticated users access to TELNET on Policy Director server truffle.

Controlling Access to Application Servers

To control access to an application (non-Policy Director) server from a NetSEAT client or a NetSEAL server, you must first create the network (or individual nodes). Then you must create a port on the network.

```
ivadmin> netseal network add 10.132.0.0 255.255.0.0 labrador

ivadmin> netseal port add labrador telnet

ivadmin> acl attach /NetSEAL/labrador/telnet auth-access
```

The above commands will permit an authenticated user to access TELNET on any node in the subnet specified by labrador. For example, a NetSEAT client will be allowed to TELNET to a node with the address 10.132.45.24.

The commands:

```
ivadmin> netseal network add 10.132.45.24 255.255.255.255 chocolate

ivadmin> netseal port add chocolate telnet

ivadmin> acl attach /NetSEAL/chocolate/telnet auth-access
```

will allow an authenticated user to TELNET to the address 10.132.45.24. In this configuration, the ACL attached to chocolate will be checked rather than the ACL attached to labrador.

Controlling Access to a Policy Director Server

A user on a node in a network protected by a Policy Director server can use TELNET to log into any node on that network. Because the Policy Director server provides the access control for users outside of the network, you may not want to allow TELNET access to the Policy Director server. When TELNETing from a node on the trusted network into the Policy Director server, the user will not be authenticated.

The commands:

```
ivadmin> netseal port add truffle telnet

ivadmin> acl attach /NetSEAL/truffle/telnet auth-access
```

will prevent the network user from TELNETing to the Policy Director server truffle (the

unauthenticated ACL entry contains no connect (C) permission).

Controlling Access from a Policy Director Server

Currently, NetSEAL cannot access an authenticated user's credentials to perform ACL checks when trapping outgoing services. All ACL checks are performed against the unauthenticated user.

The commands:

```
ivadmin> netseal network add 10.0.0.0 0.0.0.0 whole-network

ivadmin> netseal port add whole-network telnet

ivadmin> acl attach /NetSEAL/whole-network/telnet auth-access
```

will prevent an interactive client on the Policy Director server from TELNETing to any node on the network specified by the alias whole-network. Note that the netmask — made up of all zeroes — will render the IP address irrelevant.

Note: Because this is a distributed port configuration, all servers in the domain will be affected.

Controlling Access to a NetSEAL Junction

If you wish to establish a secure tunnel through an untrusted network between two Policy Director servers, then you can create a NetSEAL junction between the two Policy Director servers.

Traffic to networks nodes on the other side of the junction are routed through a secure tunnel instead of being routed over TCP. Users accessing nodes across a junction must have an ACL entry containing the forward (f) permission.

The commands:

```
ivadmin> acl modify auth-access set junction-users TCf

ivadmin> netseal junction add truffle <pd-server>
```

will create a junction from truffle to the Policy Director server specified by the *pd-server* argument. The second Policy Director server automatically “discovers” the network it is protecting at start time.

Appendix A. Policy Director administration using ivadmin

The following sections in Appendix A contain additions and corrections:

Introducing the ivadmin utility

Add the following new section:

Using ivadmin in an LDAP environment

The **ivadmin user**, **group**, **rsrc**, and **policy** commands are only appropriate in an LDAP environment. If you attempt to use these commands in a DCE environment, error messages will appear.

Using the ivadmin commands

Server commands

server flushlogs

The following **ivadmin** command:

`server flush_logs` *`server-name`*

should read as follows:

`server flushlogs` *`server-name`*

server modify baseurl

The description of this command should read as follows:

server modify <*server-name*> baseurl <*base-url*>	
	Specifies the branch of the ACL space to be used by this server. For use with replicated WebSEAL servers. Designates the specified branch ***<base-url>*** to be the master branch used by the Console for ACL administration. ACLs in this branch are applied to all replicated servers junctioned to this mount point; the replicated servers immediately reflect all ACL changes. Note that ***<base-url>*** is relative to the /WebSEAL container object, and must be located in the WebSEAL directory (not in any subdirectories).

User management commands

user list-gsouser

This command is missing from the current documentation:

user list-gsouser ***<pattern> <max-return>***	

	Generates a list of only the GSO users, listed by distinguished names. The list displays in the order the GSO users were created.
	
	See **user list** command above for details on the command arguments.
	
	Example:
	`ivadmin> user list-gsouser *luca* 2`
	
	Would produce a list similar to:
	`cn=Diana Lucas,ou=Austin,o=Wesley, Inc,c=US`
	`cn=Mike Lucaser,ou=Austin,o=Wesley, Inc,c=US`

registry policy management commands

WebSEAL only supports (enforces) the following three **ivadmin policy** commands:

```
policy {set|get} account-expiry-date
policy {set|get} max-password-age
policy {set|get} min-password-length
```

Documentation for:

policy set min-password-length [*number*]	
policy get min-password-length	

	Manages the policy for the minimum length, in characters, for a password. The *number* argument is the minimal length allowed for a password.
	
	Example:
	`ivadmin> policy set min-password-length 8`
	
	Or:
	`ivadmin> policy get min-password-length`

The following documented **ivadmin policy** commands are no longer valid:

```
policy {set|get} password-expiry-date
policy {set|get} max-account-age
```

The following documented **ivadmin** command:

```
policy {set|get} password-expiry-warn
```

has been changed to:

```
policy {set|get} number-warn-days
```

The **-user** option for the **ivadmin policy** commands is only appropriate for the **policy account-expiry-date** command.

Appendix B. 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 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
North Castle Drive
Armonk, NY 10504-1785
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 information. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this information at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

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 LZKS
11400 Burnet Road
Austin, TX 78758
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 document 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.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written.

These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (*your company name*) (*year*). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. *enter the year or years*. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, or other countries, or both:

AIX
IBM
FirstSecure
Policy Director
SecureWay

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

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

UNIX is a registered trademark in the United States, other countries, or both and is licensed exclusively through X/Open Company Limited.

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

AuthAPI	DASCOM, Inc.
DASCOM	DASCOM, Inc.
IntraVerse	DASCOM, Inc.
Internet Information Server (IIS)	Microsoft Corporation
Internet Explorer	Microsoft Corporation
Netscape	Netscape Communications Corporation
Netscape Communicator	Netscape Communications Corporation
Netscape logos	Netscape Communications Corporation
Netscape FastTrack	Netscape Communications Corporation
Netscape Navigator	Netscape Communications Corporation
NetSEAL	DASCOM, Inc.

NetSEAT	DASCOM, Inc.
Smart Junctions	DASCOM, Inc.
Solaris	Sun Microsystems, Inc
WebSEAL	DASCOM, Inc.