IBM Security Verify Access
Version 10.0.8
June 2024

Web Reverse Proxy stanza reference topics

Contents

Tables

Chapter 1. Stanza reference

This guide provides a complete stanza reference for the WebSEAL configuration file, alphabetized by stanza name.

You can use the appliance Local Management Interface (LMI) to edit the WebSEAL configuration file. On the Reverse Proxy management page, select the appropriate WebSEAL instance and click **Manage** > **Configuration** > **Edit Configuration File** to open the Advanced Configuration File Editor. You can use this editor to directly edit the WebSEAL configuration file.

[acnt-mgt] stanza

Use the **[acnt-mgt]** stanza to configure the WebSEAL account management pages.

account-expiry-notification

Use the **account-expiry-notification** stanza entry to control how WebSEAL reports login failures that are caused by invalid or expired accounts.

Syntax

```
account-expiry-notification = {yes|no}
```

Description

Specifies whether WebSEAL informs the user of the reason for a login failure when the failure is caused by an invalid or expired account. When this entry is set to no, the user receives the same error message as the message that is sent when a login fails as a result of invalid authentication information, such as an invalid user name or password.

Options

yes
Enable.

no
Disable.

Usage

This stanza entry is required.

Default value

no

Example

```
account-expiry-notification = yes
```

account-inactivated

Use the **account-inactivated** stanza entry to configure the page that WebSEAL displays when a user with an inactive account tries to log in with the correct password.

Syntax

```
account-inactivated = filename
```

Description

Page that is displayed when nsAccountLock is true for a user (in Sun Directory Server) when they attempt to log in. WebSEAL displays the specified page only if the user provides the correct password during login.

Note: This option has no effect unless the corresponding Security Verify Access LDAP option is enabled (**[ldap] enhanced-pwd-policy**=yes). This LDAP option must be supported for the particular LDAP registry type.

Options

filename
Page that is displayed when nsAccountLock is true for the user who provides the correct password during login.

Usage

This stanza entry is required.

Default value

None.

Note: The value for this option in the template configuration file is acct_locked.html.

Example

```
account-inactivated = acct_locked.html
```

account-locked

Use the **account-locked** stanza entry to configure the page that WebSEAL displays when a user authentication fails because the account is locked.

Syntax

```
account-locked = filename
```

Description

Page that is displayed when the user authentication fails as a result of a locked user account.

Options

filename
Page that is displayed when the user authentication fails as a result of a locked user account.

Usage

This stanza entry is required.

Default value

`acct_locked.html`

Example

```
account-locked = acct_locked.html
```

allow-unauthenticated-logout

Use the **allow-unauthenticated-logout** stanza entry to control whether unauthenticated users can request the pkmslogout resource.

Syntax

```
allow-unauthenticated-logout = {yes|no}
```

Description

Determines whether unauthenticated users are able to request the pkmslogout resource without authenticating first.

Options

yes
Unauthenticated users can request the pkmslogout resource.

no
Unauthenticated users must authenticate before the pkmslogout resource is returned.

Usage

This stanza entry is required.

Default value

no

Example

```
allow-unauthenticated-logout = no
```

allowed-referers

Use the **allowed-referers** stanza entry to specify which referrers can request management pages.

Syntax

```
allowed-referers = referer_filter
```

Description

For protection against cross-site request forgery (CSRF) attacks, you can configure WebSEAL to validate the HTTP Request **referer** header for all account management pages. WebSEAL uses the value that is provided for this configuration entry to determine whether the referrer host name in an incoming request is "valid".

If this entry is configured, when WebSEAL receives a request for an account management page, WebSEAL:

1. Checks whether the **referer** header is present in the HTTP Request header.
2. Validates the host name portion of that referrer against the **allowed-referers** entries.

If WebSEAL finds that an incoming request does not match any of the configured **allowed-referers** filters, the request fails and WebSEAL returns an error page.

Entries can contain the following wildcard characters:

- * - match 0 or more characters.
- ? - match any single character.
- \ - Literal match of the following character.

You can use the value %HOST% for this entry. This value is a special filter, which indicates to WebSEAL that a referrer is "valid" if the host name portion of the **referer** header matches the **host** header.

If there are no **allowed-referers** entries then WebSEAL does not complete this validation.

Note: You can specify this entry multiple times to define multiple "allowed" referrer filters. WebSEAL uses all of these entries to validate the referrer.

For more information about referrer validation, search for "CSRF" in the *IBM Security Verify Access: Web Reverse Proxy Configuration Guide*.

Options

referer_filter
Specifies a filter for a referrer host name that WebSEAL can accept as "valid".

Usage

This stanza entry is optional.

Default value

None.

Example

The following entry matches any referrer host name that begins with the characters `ac`, followed by zero or more characters, and ends with the characters `me`.

```
allowed-referers = ac*me
```

The following entry indicates that a referrer is "valid" if the host name portion of the **referer** header matches the **host** header.

```
allowed-referers = %HOST%
```

cert-failure

Use the **cert-failure** stanza entry to specify the page that is displayed if certificates are required and a client fails to authenticate with a certificate.

Syntax

```
cert-failure = filename
```

Description

Page displayed when certificates are required and a client fails to authenticate with a certificate.

Options

filename
Page displayed when certificates are required and a client fails to authenticate with a certificate.

Usage

This stanza entry is required.

Default value

`certfailure.html`

Example

```
cert-failure = certfailure.html
```

cert-stepup-http

Use the **cert-stepup-http** stanza entry to specify the error page that WebSEAL displays if a user attempts to increase the authentication strength level to certificate authentication from an HTTP session.

Syntax

```
cert-stepup-http = filename
```

Description

WebSEAL displays this HTML page when a client attempts to increase authentication strength level (step-up) to certificates while using HTTP protocol.

Options

filename
WebSEAL displays this HTML page when a client attempts to increase authentication strength level (step-up) to certificates while using HTTP protocol.

Usage

This stanza entry is required.

Default value

`certstepuphttp.html`

Example

```
cert-stepup-http = certstepuphttp.html
```

certificate-login

Use the **certificate-login** stanza entry to specify the login request form that WebSEAL uses for client-side certificate authentication.

Syntax

```
certificate-login = filename
```

Description

Form requesting client-side certificate authentication login.

This form is used only when the **accept-client-certs** key in the **[certificate]** stanza is set to prompt_as_needed.

Options

filename
Form requesting client-side certificate authentication login.

Usage

This stanza entry is required when delayed certificate authentication or authentication strength level (step-up) for certificates is enabled.

Default value

certlogin.html

Example

```
certificate-login = certlogin.html
```

change-password-auth

Use the **change-password-auth** stanza entry to control whether the user is automatically authenticated, if required, during a change password request.

Syntax

```
change-password-auth = {yes|no}
```

Description

Enable this option to automatically authenticate users during password change operations. If the password for the user is expired, and this option is enabled, then WebSEAL completes the following tasks:

- Authenticates the user with the expired password.
- Changes the password.
- Automatically authenticates the user as required during the password change operation.

This configuration is helpful in failover situations. The user might be served the password change form from one WebSEAL replica, but the form posts to another replica where the user session does not exist.

Options

yes
Enable.

no
Disable.

Usage

This stanza entry is required.

Default value

no

Example

```
change-password-auth = yes
```

client-notify-tod

Use the **client-notify-tod** stanza entry to control whether WebSEAL displays an error page if authorization is denied as a result of a POP time of day check.

Syntax

```
client-notify-tod = {yes|no}
```

Description

Enable the display of an error page when authorization is denied due to a POP time of day check. The error page is `38cf08cc.html`.

Options

yes
Enable.

no
Disable.

Usage

This stanza entry is required.

Default value

no

Example

```
client-notify-tod = yes
```

default-response-type

Use the **default-response-type** entry to specify the response type of WebSEAL generated responses when the 'accept' and 'content-type' headers are missing from the request.

Syntax

```
default-response-type = <MIME type>/<MIME subtype>
```

Description

When you are generating a response page, WebSEAL attempts to use the template that best matches the expected response type.

It uses the 'accept' and 'content-type' HTTP request headers when determining the correct template to be used. If both headers are not present in the request, or an appropriate file could not be located, the default template response type, as defined by this configuration is used.

Different default-response types can be specified for different user agents by appending to the configuration entry name a user agent string. The '*' and '?' characters can be used when matching the user agent. For example,

```
default-response-type:*Mobile* = application/json
default-response-type = text/html
```

Options

default-response-type
Specifies the response type to be used when the 'accept' and 'content-type' headers are missing from the request.

Usage

This stanza entry is required.

Default Value

None.

Example

```
default-response-type = text/html
```

enabled-html-languages

Use the `enabled-html-languages` stanza entry to specify the languages which can be used by WebSEAL when generating error or management responses.

Syntax

```
enabled-html-languages = <language-list>
```

Description

Specifies a comma-separated list of languages which are enabled for WebSEAL generated error and management pages. The accept-language HTTP header from the request is used by WebSEAL to determine the language used when generating responses.

The first language in the configuration entry will be treated as the default language. The default language will be used if none of the languages contained in the 'accept-language' HTTP header have been enabled.

The list of supported languages, designated by Language Code, include:

```
C, cs, de, es, fr, hu, it, ja, ko, pl, pt_BR, ru, zh_CN, zh_TW
```

If there are no languages specified, all languages are automatically enabled, with English (C) set as the default.

Options

<language-list>
A comma separated list of languages which are enabled for WebSEAL generated error and management pages.

Usage

This stanza entry is optional.

Default value

None. If no value is specified, all languages are automatically enabled.

Example

```
enabled-html-languages = C,de,es
```

enable-html-redirect

Use the **enable-html-redirect** stanza entry to enable HTML redirection. You can use HTML redirection, in conjunction with some JavaScript code, to preserve the HTML fragment in the response.

Syntax

```
enable-html-redirect = {yes|no}
```

Description

Configures WebSEAL to use the HTML redirect page to handle redirections rather than returning an HTTP 302 response redirect.

When a user successfully authenticates, WebSEAL typically uses an HTTP 302 response to redirect the user back to the resource that was originally requested.

HTML redirection causes WebSEAL to send a static page back to the browser instead of a 302 redirect. WebSEAL can then use the JavaScript or any other code that is embedded in this static page to process the redirect.

You can use the **html-redirect** configuration entry, which is also in the **[acnt-mgt]** stanza, to specify the page that contains the HTML redirection.

For more information about HTML redirection, see the *IBM Security Verify Access: Web Reverse Proxy Configuration Guide*.

Note: If you enable this configuration entry, you must not specify a value for the **login-redirect-page** entry, which is also in the **[acnt-mgt]** stanza.

Options

yes
Enable.

no
Disable.

Usage

This stanza entry is required.

Default value

no

Example

```
enable-html-redirect = no
```

enable-passwd-warn

Use the **enable-passwd-warn** stanza entry to configure WebSEAL to display a password warning form when it detects the REGISTRY_PASSWORD_EXPIRE_TIME attribute in the user credential at login. This attribute indicates that the user password is soon to expire.

Syntax

```
enable-passwd-warn = {yes|no}
```

Description

Enable WebSEAL to detect the attribute REGISTRY_PASSWORD_EXPIRE_TIME added to a users' credential when the LDAP password policy indicates that their password is soon to expire. The value of this attribute is the number of seconds until their password expires. When this attribute is detected, at login to WebSEAL, a password warning form will appear.

NOTE: This option must be set in order to use the associated options, which are also in the **[acnt-mgt]** stanza: **passwd-warn** and **passwd-warn-failure**. The corresponding Security Verify Access LDAP option must be enabled (`[ldap] enhanced-pwd-policy=yes`) and supported for the particular LDAP registry type.

Options

yes
Enable the detection of the REGISTRY_PASSWORD_EXPIRE_TIME to ultimately warn the user when their password is soon to expire.

no
Disable the detection of the REGISTRY_PASSWORD_EXPIRE_TIME attribute. WebSEAL will not be able to notify users when their passwords are soon to expire.

Usage

This stanza entry is optional.

Default value

The option will default to `yes` if it is not specified in the configuration file.

NOTE: The value for this option in the template configuration file is no.

Example

```
enable-passwd-warn = yes
```

enable-secret-token-validation

Use the **enable-secret-token-validation** stanza entry to enable secret token validation, which protects certain WebSEAL account management pages against cross-site request forgery (CSRF) attacks.

Syntax

```
enable-secret-token-validation = {true|false}
```

Description

Use this entry to enable secret token validation, which protects certain WebSEAL account management pages against cross-site request forgery (CSRF) attacks. If you set this entry to `true`, WebSEAL adds a token to each session and validates the "token" query argument for the following account management requests:

- /pkmslogin.form
- /pkmslogout
- /pkmslogout-nomas
- /pkmssu.form
- /pkmsskip
- /pkmsdisplace
- /pkmspaswd.form
- /pkmsoidc

For example, you must change the `/pkmslogout` request to `pkmslogout?token=<value>`, where `<value>` is the unique session token.

If secret token validation is enabled and the token argument is missing from the request or does not match the session token, WebSEAL returns an error page. For more information about secret token validation, search for "CSRF" in the *IBM Security Verify Access: Web Reverse Proxy Configuration Guide*.

Options

`true`
WebSEAL uses secret token validation to protect against CSRF attacks.

Note: This setting modifies the URLs for the affected WebSEAL management pages. Each of these management requests must contain a "token" argument with the current session token.

`false`
WebSEAL does not use secret token validation.

Usage

This stanza entry is optional.

Default value

`false`

Example

```
enable-secret-token-validation = true
```

help

Syntax

```
help = filename
```

Description

Page containing links to valid administration pages.

Options

filename
Page containing links to valid administration pages.

Usage

This stanza entry is required.

Default value

`help.html`

Example

```
help = help.html
```

http-rsp-charset

Use the `http-rsp-charset` stanza entry to define the character set which will be included in the content-type HTTP header for WebSEAL generated responses.

Syntax

```
http-rsp-charset = character-set
```

Description

Specifies the character set which will be included in the content-type HTTP header for WebSEAL generated response pages.

Options

character-set
The character set to be included.

Usage

The stanza entry is optional.

Default Value

None.

Example

```
http-rsp-charset = UTF-8
```

http-rsp-header

Syntax

```
http-rsp-header = header-name:macro
```

Description

Inserts custom headers whenever WebSEAL returns a custom response to the client.

Options

header-name
The name of the header that holds the value.

macro
That type of value to be inserted. This parameter can be one of the following values:

- TAM_OP
- AUTHNLEVEL
- ERROR_CODE
- ERROR_TEXT
- CREDATTR{*<name>*}, where *<name>* is the name of the credential attribute.
- USERNAME
- TEXT{*<value>*}, where *<value>* is the static header to include in the response header.

Usage

This stanza entry is optional.

Note: You can specify this entry multiple times to include multiple headers in the response.

Default value

None.

Example

The following example inserts the Security Verify Access error code in a response header named `tam-error-code`:

```
http-rsp-header = tam-error-code:ERROR_CODE
```

The following example includes a static header DENY in a response header named `X-Frame-Options`:

```
http-rsp-header = x-frame-options:TEXT{DENY}
```

html-redirect

Syntax

```
html-redirect = filename
```

Description

Specifies the standard HTML redirection page.

Options

filename
Standard HTML redirection page.

Usage

This stanza entry is required.

Default value

`redirect.html`.

Example

```
html-redirect = redirect.html
```

login

Syntax

```
login = filename
```

Description

Standard login form.

Options

filename
Standard login form.

Usage

This stanza entry is required.

Default value

`login.html`

Example

```
login = login.html
```

login-redirect-page

Syntax

```
login-redirect-page = destination
```

Description

Page to which users are automatically redirected after completing a successful authentication. The configured redirect destination can be either:

- A server-relative Uniform Resource Locator (URL), or
- An absolute URL, or
- A macro which allows dynamic substitution of information from WebSEAL.

The supported macros include:

%AUTHNLEVEL%
Level at which the session is currently authenticated.

%HOSTNAME%
Fully qualified host name.

%PROTOCOL%
The client connection protocol used. Can be HTTP or HTTPS.

%URL%
The original URL requested by the client.

%USERNAME%
The name of the logged in user.

%HTTPHDR{*name*}%
The HTTP header that corresponds to the specified name. For example: `%HTTPHDR{Host}%`

%CREDATTR{*name*}%
The credential attribute with the specified name. For example: `%CREDATTR{tagvalue_session_index}%`

Note: You cannot use this configuration entry if the **enable-html-redirect** entry (also in the **[acnt-mgt]** stanza) is set to `yes`. These redirects are not compatible with one another.

In order for the configured login redirect to take effect, the redirect capability **must** be enabled for the desired authentication mechanisms by using the "[enable-redirects] stanza" on page 98.

Options

destination
Uniform Resource Locator (URL) to which users are automatically redirected after login, or a macro for dynamic substitution of information from WebSEAL.

Usage

This stanza entry is optional.

Default value

None.

Example

Example of a server relative URL:

```
login-redirect-page = /jct/page.html
```

Example of an absolute URL:

```
login-redirect-page = http://www.ibm.com/
```

Example that uses a macro:

```
login-redirect-page = /jct/intro-page.html?level=%AUTHNLEVEL%&url=%URL%
```

login-success

Syntax

```
login-success = filename
```

Description

Page displayed after successful login.

Options

filename
Page displayed after successful login.

Usage

This stanza entry is required.

Default value

`login_success.html`

Example

```
login-success = login_success.html
```

logout

Syntax

```
logout = filename
```

Description

Page displayed after successful logout.

Options

filename
Page displayed after successful logout.

Usage

This stanza entry is required.

Default value

`logout.html`

Example

```
logout = logout.html
```

oidc-fragment

Use this entry to define the page to be displayed during an OIDC implicit authentication flow.

Syntax

```
oidc-fragment = filename
```

Description

Page to be displayed during an OIDC implicit authentication flow.

Options

filename
File name of the page to be displayed during an OIDC implicit authentication flow.

Usage

This stanza entry is required.

Default value

`oidc_fragment.html`

Example

```
oidc-fragment = oidc_fragment.html
```

passwd-change

Syntax

```
passwd-change = filename
```

Description

Page containing a change password form.

Options

filename
Page containing a change password form.

Usage

This stanza entry is required.

Default value

`passwd.html`

Example

```
passwd-change = passwd.html
```

passwd-change-failure

Syntax

```
passwd-change-failure = filename
```

Description

Page displayed when password change request fails.

Options

filename
Page displayed when password change request fails.

Usage

This stanza entry is required.

Default value

`passwd.html`

Example

```
passwd-change-failure = passwd.html
```

passwd-change-success

Syntax

```
passwd-change-success = filename
```

Description

Page displayed when password change request succeeds.

Options

filename
Page displayed when password change request succeeds.

Usage

This stanza entry is required.

Default value

`passwd_rep.html`

Example

```
passwd-change-success = passwd_rep.html
```

passwd-expired

Syntax

```
passwd-expired = filename
```

Description

Page displayed when the user authentication fails due to an expired user password.

Options

filename
Page displayed when the user authentication fails due to an expired user password.

Usage

This stanza entry is required.

Default value

`passwd_exp.html`

Example

```
passwd-expired = passwd_exp.html
```

passwd-warn

Syntax

```
passwd-warn = filename
```

Description

Page displayed after login if WebSEAL detects the LDAP password is soon to expire.

NOTE: This option has no effect unless **enable-passwd-warn** (also in the **[acnt-mgt]** stanza) is set to `yes` and the corresponding Security Verify Access LDAP option is also enabled (`[ldap] enhanced-pwd-policy=yes`). This LDAP option must be supported for the particular LDAP registry type.

Options

filename
Page displayed as a warning that the LDAP password is soon to expire.

Usage

This stanza entry is required.

Default value

None.

NOTE: The value for this option in the template configuration file is `passwd_warn.html`.

Example

```
passwd-warn = passwd_warn.html
```

passwd-warn-failure

Syntax

```
passwd-warn-failure = filename
```

Description

Page displayed if the user fails to change their password after being notified that the LDAP password is soon to expire. This page gives the user another chance to change their password and indicates the cause of the error.

NOTE: This option has no effect unless **enable-passwd-warn** (also in the **[acnt-mgt]** stanza) is set to `yes` and the corresponding Security Verify Access LDAP option is also enabled (`[ldap] enhanced-pwd-policy=yes`). This LDAP option must be supported for the particular LDAP registry type.

Options

filename
Page displayed if the user does not change their password after receiving notification that the LDAP password is soon to expire.

Usage

This stanza entry is required.

Default value

None.

NOTE: The value for this option in the template configuration file is `passwd_warn.html`.

Example

```
passwd-warn-failure = passwd_warn.html
```

pkmspublic-uri

Use this entry to define the URI to be used when accessing the custom template pages.

Syntax

```
pkmspublic-uri = uri
```

Description

The single path segment which is used to reference the custom template files.

Note: The single path segment should not contain a '/' character.

The custom template files can be accessed at the following URI: /<pkmspublic-uri>. For example: /pkmspublic/background.png.

Options

uri
The URI to be used when accessing the custom template pages.

Usage

This stanza entry is optional.

Default value

pkmspublic

Example

```
pkmspublic-uri = custom
```

single-signoff-uri

Syntax

```
single-signoff-uri = URI
```

Description

When a user session is terminated in WebSEAL, any sessions that might exist on backend application servers are not destroyed. You can use this configuration entry to change this default behavior.

When a WebSEAL user session is terminated and this stanza entry is configured, WebSEAL sends a request to the resource specified by the configured URI. The request contains any configured headers and cookies for the junction point on which the resource resides. The backend application can use this information to terminate any sessions for that user.

Note: You can configure more than one **single-sign-off-uri** entry to send a request to multiple URIs.

Options

URI

The resource identifier of the application that receives the single signoff request from WebSEAL.

Note: The URI must be server relative and correspond to a resource on a standard junction.

Usage

This stanza entry is optional.

Default value

None.

Example

```
single-signoff-uri = /management/logoff
```

stepup-login

Syntax

```
stepup-login = filename
```

Description

Step-up authentication login form.

Options

filename
Step-up authentication login form.

Usage

This stanza entry is required.

Default value

`stepuplogin.html`

Example

```
stepup-login = stepuplogin.html
```

switch-user

Syntax

```
switch-user = filename
```

Description

Switch user management form.

Options

filename
Switch user management form.

Usage

This stanza entry is required.

Default value

`switchuser.html`

Example

```
switch-user = switchuser.html
```

temp-cache-response

Syntax

```
temp-cache-response = filename
```

Description

The default page that WebSEAL returns if no URL redirect is supplied with the `pkmstempsession` request. The `pkmstempsession` page is accessed to achieve session sharing with Microsoft Office applications. For more information about sharing sessions with Microsoft Office applications, see the *IBM Security Verify Access: Web Reverse Proxy Configuration Guide*.

Options

filename
The default page that WebSEAL returns for a `pkmstempsession` request.

Usage

This stanza entry is optional.

Default value

`temp_cache_response.html`

Example

```
temp-cache-response = temp_cache_response.html
```

too-many-sessions

Syntax

```
too-many-sessions = filename
```

Description

Page displayed when a user has too many concurrent sessions and must either cancel their new login or terminate the other sessions.

Options

filename
Page displayed when a user has too many concurrent sessions and must either cancel their new login or terminate the other sessions.

Usage

This stanza entry is required.

Default value

`too_many_sessions.html`

Example

```
too-many-sessions = too_many_sessions.html
```

use-restrictive-logout-filenames

Syntax

```
use-restrictive-logout-filenames = {yes|no}
```

Description

Control the restrictions normally enforced on the name of the /pkmslogout custom response file.

Options

yes
Use default restrictions to enforce the name of the /pkmslogout custom response file.

no
Only slash (/), backslash (\), characters outside of the ASCII range 0x20 - 0x7E, and filenames that begin with a period (.) will be disallowed.

Usage

This stanza entry is required.

Default value

`yes`

Example

```
use-restrictive-logout-filenames = yes
```

use-filename-for-pkmslogout

Syntax

```
use-filename-for-pkmslogout = {yes|no}
```

Description

Controls whether or not the appended query string (specifying a custom response page) in a **pkmslogout** command is used to override the default response page.

Options

yes
Enables the operation of the query string. If a query string in a **pkmslogout** URL specifies a custom response page, that custom page is used instead of the default page.

no
Disables the operation of the query string. Any query string in a **pkmslogout** URL that specifies a custom response page is ignored. Only the default response page is used upon logout.

Usage

This stanza entry is required.

Default value

no

Example

```
use-filename-for-pkmslogout = yes
```

[acnt-mgt:<jct-id>] stanza

enable-local-response-redirect

Use the **enable-local-response-redirect** stanza entry to enable or disable local response redirection. When local response redirection is enabled, the redirection is used for all local WebSEAL response types: login, error, informational, and password management.

Syntax

```
enable-local-response-redirect = {yes|no}
```

Description

Enable or disable sending a redirection to a response application instead of serving management or error pages from the local system.

You can customize this configuration item for a particular junction by adding the adjusted configuration item to a **[acnt-mgt:*{junction_name}*]** stanza.

where *{junction_name}* refers to the junction point for a standard junction (including the leading / character) or the virtual host label for a virtual host junction.

Options

yes
Enable.

no
Disable.

Usage

This stanza entry is required.

Default value

no

Example

```
enable-local-response-redirect = no
```

[authentication-levels] stanza

Use the **[authentication-levels]** stanza to define the step-up authentication levels.

level

Syntax

```
level = method-name
```

Description

Step-up authentication levels. WebSEAL enables authenticated users to increase the authentication level by use of step-up authentication. This *key=value* pair specifies which step-up authentication levels are supported by this WebSEAL server.

Do not specify an authentication level unless the authentication method is enabled. For example, you must enable either basic authentication or forms authentication before you set `level = password`.

Enter a separate *key=value* pair for each supported level. Supported levels include:

- ext-auth-interface
- ltpa
- oidc
- password
- ssl
- unauthenticated

The position of the entry in the file dictates the associated authentication level. The first row, typically `unauthenticated`, is associated with authentication level of `0`. Each subsequent line is associated with the next higher level. You can add multiple entries for the same method.

It is possible for the method to set the authentication level itself. For example, an External Authentication Interface (EAI) implementation might set either authentication level of 2 or 3 depending on the authentication transaction that the client undertakes.

The EAI can set this authentication level directly in the identity attributes returned to WebSEAL. To support this implementation, you can create two identical lines in positions 3 and 4. For example:

```
level = unauthenticated         (associated with level 0)
level = password                        (associated with level 1)
level = ext-auth-interface    (associated with level 2)
level = ext-auth-interface  (associated with level 3)
```

Options

method-name
Name of authentication method.

Usage

This stanza entry is required.

Default value

`unauthenticated`

`password`

Example

```
level = unauthenticated
level = password
```

[aznapi-configuration] stanza

Use the **[aznapi-configuration]** stanza to configure the authorization API services.

audit-attribute

Use the **audit-attribute** stanza entry to list the attributes to audit.

Syntax

```
audit-attribute = attribute
```

Description

Attributes to be audited.

Note: You can configure multiple **audit-attribute** entries. Create a separate **audit-attribute** entry for each attribute to be audited.

Options

attribute
Attributes to be audited.

Usage

This stanza entry is required.

Default value

`tagvalue_su-admin`

Example

```
audit-attribute =   tagvalue_su-admin
```

auditcfg

Use the **auditcfg** stanza entry to configure which events WebSEAL audits.

Syntax

```
auditcfg = {azn|authn|http}
```

Description

Indicates the components for which auditing of events is configured. To enable component-specific audit records, add the appropriate definition.

Options

azn
Capture authorization events.

authn
Capture authentication events.

http
Capture HTTP events. These events correspond to the events logged by the request, referrer, and agent logging clients.

Usage

This stanza entry is optional for WebSEAL. However, this stanza entry is required when auditing is enabled (`logaudit = yes`).

Default value

There is no default value for WebSEAL because auditing is disabled by default.

Example

Create a separate stanza entry for each component to be activated. The components are included in the default configuration file, but are commented out. To activate a commented out entry, remove the pound sign (#) from the start of the entry.

Example:

```
auditcfg = azn
#auditcfg = authn
#auditcfg = http
```

audit-json

Use the `audit-json` entry to control whether the auditing records are written in JSON format.

Syntax

```
audit-json = {yes|no}
```

Description

This configuration entry determines whether auditing records, including the request log, are formatted as XML or JSON.

Options

yes
Auditing records are formatted as JSON.

no
Auditing records are formatted as XML.

Usage

This stanza entry is optional.

Default Value

no

Example

```
audit-json = yes
```

cache-refresh-interval

Use the **cache-refresh-interval** stanza entry to specify the interval in seconds between polls to the master authorization server to check for updates.

Syntax

```
cache-refresh-interval = {disable|default|number_of_seconds}
```

Description

Poll interval between checks for updates to the master authorization server.

Options

disable
The interval value in seconds is not set.

default
When value is to `default`, an interval of 600 seconds is used.

number_of_seconds
Integer value indicating the number of seconds between polls to the master authorization server to check for updates.

The minimum number of seconds is 0. There is no maximum value.

Usage

This stanza entry is optional.

Default value

`disable`

Example

```
cache-refresh-interval = disable
```

client-ip-http-header

Use this entry to specify the header which contains the client IP address.

Syntax

```
client-ip-http-header = <header-name>
```

Description

This configuration entry is used to define the name of the HTTP header which contains the IP address of the client. This IP address will be used as the client address in authorization decisions and auditing records. If no HTTP header is configured, or the configured HTTP header is missing from the HTTP request, or the contents of the HTTP header does not correspond to an IP address, the client IP address of the connection itself will be used instead.

Options

<header-name>
The name of the HTTP header which contains the client IP address.

Usage

This stanza entry is optional.

Default value

None

Example

```
client-ip-http-header = X-Forwarded-For
```

input-adi-xml-prolog

Syntax

```
input-adi-xml-prolog = prolog
```

Description

The prolog to be added to the top of the XML document that is created using the Authorization Decision Information (ADI) needed to evaluate a boolean authorization rule.

Options

prolog
The prolog to be added to the top of the XML document that is created using the Authorization Decision Information (ADI) needed to evaluate a boolean authorization rule.

Usage

This stanza entry is optional.

Default value

<?xml version='1.0' encoding='UTF-8'?>

Example

```
input-adi-xml-prolog = <?xml version='1.0' encoding='UTF-8'?>
```

listen-flags

Syntax

```
listen-flags = {enable|disable}
```

Description

Enables or disables the reception by WebSEAL of policy cache update notifications from the master authorization server.

Options

enable
Activates the notification listener.

disable
Deactivates the notification listener.

Usage

This stanza entry is required.

Default value

disable

Example

```
listen-flags = enable
```

logaudit

Syntax

```
logaudit = {yes|true|no|false}
```

Description

Enables or disables auditing.

Options

yes
Enable auditing.

true
Enable auditing.

no
Disable auditing.

false
Disable auditing.

Usage

This stanza entry is required.

Default value

no

Example

```
logaudit = no
```

logclientid

Syntax

```
logclientid = webseald
```

Description

Name of the daemon whose activities are audited through use of authorization API logging.

Options

webseald
Name of the daemon whose activities are audited through use of authorization API logging.

Usage

This stanza entry is required.

Default value

webseald

Example

```
logclientid = webseald
```

logcfg

Syntax

```
logcfg = category:{stdout|stderr|file|remote|rsyslog}[ [parameter=value ]
[,parameter=value]...]
```

Description

Specifies event logging for the specified *category*.

Options

Specifies event logging for the specified *category*.

For WebSEAL, the categories are:

audit.azn
Authorization events.

audit.authn
Credentials acquisition authentication.

http
All HTTP logging information.

http.clf
HTTP request information as defined by the request-log-format configuration entry in the **`[logging]`** stanza.

http.ref
HTTP Referer header information.

http.agent
HTTP User_Agent header information

{stdout|stderr|file|remote|rsyslog}
Event logging supports a number of output destination types. WebSEAL auditing typically is configured to use the `file` type.

parameter = value

Each event logging type supports a number of optional *`parameter = value`* options.

For more information about output destination types and optional *parameter = value* settings, see the *IBM Security Verify Access for Web: Administration Guide*.

Usage

This stanza entry is optional.

Default value

None.

Example

Example entry for `request.log` (common log format) (entered as one line):

```
logcfg = http.clf:file path=request_file,
flush=time,rollover_size=max_size,
max_rollover_files=max_files,log_id=httpclf,buffer_size=8192,
queue_size=48
```

logflush

Syntax

```
logflush = number_of_seconds
```

Description

Integer value indicating the frequency, in seconds, to force a flush of log buffers.

Options

number_of_seconds

The minimum value is 1 second.

The maximum value is 600 seconds.

Usage

This stanza entry is optional.

Default value

20

Example

```
logflush = 20
```

logsize

Syntax

```
logsize = number_of_bytes
```

Description

Integer value indicating the size limit of audit log files. The size limit is also referred to as the *rollover threshold*. When the audit log file reaches this threshold, the original audit log file is renamed and a new log file with the original name will be created.

Options

number_of_bytes

When the value is zero (0), no rollover log file is created.

When the value is a negative integer, the logs are rolled over daily, regardless of the size.

When the value is a positive integer, the value indicates the maximum size, in bytes, of the audit log file before the rollover occurs. The allowable range is from 1 byte to 2 megabytes

Usage

This stanza entry is optional.

Default value

2000000

Example

```
logsize = 2000000
```

permission-info-returned

Syntax

```
permission-info-returned = permission-attribute
```

Description

Specifies the permission information returned to the resource manager (for example, WebSEAL) from the authorization service.

Options

permission-attribute

The **azn_perminfo_rules_adi_request** setting allows the authorization service to request ADI from the current WebSEAL client request. The **azn_perminfo_reason_rule_failed** setting specifies that rule failure reasons be returned to the resource manager (this setting is required for –R junctions).

To enable the Privacy Redirection capabilities of the AMWebARS Web Service, the **azn_perminfo_amwebars_redirect_url** must be included.

Usage

This stanza entry is optional.

Default value

`azn_perminfo_rules_adi_request azn_perminfo_reason_rule_failed`

Example

```
permission-info-returned = azn_perminfo_rules_adi_request
azn_perminfo_reason_rule_failed
```

policy-attr-separator

Syntax

```
policy-attr-separator = separator
```

Description

Specifies the character that WebSEAL uses for the following services:

- Credential policy entitlements service.
- Registry entitlements service.

Note: For the credential policy entitlements service to work properly, a user's DN cannot contain the specified separator. If the user DN contains this separator then WebSEAL fails when attempting to retrieve the user's policy attributes.

Options

separator

The character that WebSEAL uses for the credential policy entitlements service and the registry entitlements service. Ensure that the chosen character is not present in any User DN values.

Usage

This stanza entry is optional.

Default value

By default, WebSEAL uses colon (`:`) as the separator for these services.

Example

```
policy-attr-separator = #
```

policy-cache-size

Syntax

```
policy-cache-size = cache_size
```

Description

The maximum size of the in-memory policy cache is configurable. The cache consists of policy and the relationships between policy and resources. The knowledge that a resource has no directly associated policy is also cached.

Options

cache_size

The maximum cache size should be relative to the number of policy objects defined and the number of resources protected and the available memory.

A reasonable algorithm to begin with is: (number of policy objects * 3) + (number of protected resources * 3)

This value controls how much information is cached. A larger cache will potentially improve the application performance but use additional memory as well.

Size is specified as the number of entries.

Usage

This stanza entry is optional.

Default value

None.

Example

```
policy-cache-size = 32768
```

resource-manager-provided-adi

Syntax

```
resource-manager-provided-adi = prefix
```

Description

A list of string prefixes that identify Authorization Decision Information (ADI) to be supplied by the resource manager (in this case, WebSEAL).

Options

prefix

The default settings below tell the authorization engine that when it requires ADI with the prefixes AMWS_hd_, AMWS_qs_ ,or AMWS_pb_ to evaluate a boolean authorization rule, and the ADI is not available in either the credential or application context passed in with the access decision call, that the engine should fail the access decision and request that the resource manager retry the request and provide the required data in the application context of the next request.

Usage

This stanza entry is optional.

Default value

AMWS_hd_, AMWS_pb_, AMWS_qs_

Example

```
resource-manager-provided-adi = AMWS_hd_
resource-manager-provided-adi = AMWS_pb_
resource-manager-provided-adi = AMWS_qs_
```

skip-eas-on-bypass-pop

Use the **skip-eas-on-bypass-pop** entry to control whether the **BypassPop** setting is ignored if an EAS is to be called.

Syntax

```
skip-eas-on-bypass-pop = {yes|no}
```

Description

This configuration entry takes effect only when the authenticated user's effective ACL to the requested resource has the **BypassPOP** flag turned on.

Note: Setting this configuration entry to yes might affect normal EAS usage that makes authorization decisions for the access of protected resources. For example, if the **skip-eas-on-bypass-pop** entry is set to no or not present, the **BypassPOP** flag is ignored and the EAS is called for authorization when a protected resource is accessed. If this entry is set to yes, the **BypassPOP** flag takes effect and the EAS is no longer called to make the authorization decision. Take this fact into consideration when you set this configuration entry.

Options

yes
Do not ignore the **BypassPop** flag when the system decides whether to call into the EAS.

no
Ignore the **BypassPOP** flag when the system decides whether to call into the EAS.

Usage

This stanza entry is optional. If this entry is not present, the **BypassPOP** flag is ignored when the system decides whether to call into EAS.

Default value

no

Example

```
skip-eas-on-bypass-pop = no
```

special-eas

Use the `special-eas` stanza entry to allow the path to the effective POP for the protected resource being accessed, to be passed to the EAS.

Syntax

```
special-eas = EAS-trigger
```

Description

Adding the `special-eas` entry to the `[aznapi-configuration]` stanza activates the process to pass the path of the effective POP to EAS. Setting this value is necessary for multiple policy support.

Options

EAS-trigger
Specify the value that is set by the policy trigger defined in the `[aznapi-external-authzn-services]` stanza.

Usage

This stanza entry is not required.

Default value

None.

Example

If the following `rba_pop_trigger` trigger is defined:

```
[aznapi-external-authzn-services]
rba_pop_trigger = /opt/pdweb/librtsseas.so
```

Then, use the following stanza and entry to indicate that the `rba_pop_trigger` is a special trigger:

```
[aznapi-configuration]
special-eas = rba_pop_trigger
```

xsl-stylesheet-prolog

Syntax

```
xsl-stylesheet-prolog = prolog
```

Description

The prolog to be added to the top of the XSL stylesheet that is created using the XSL text that defines a boolean authorization rule.

Options

prolog
The prolog to be added to the top of the XSL stylesheet that is created using the XSL text that defines a boolean authorization rule.

Usage

This stanza entry is optional.

Default value

```
<?xml version='1.0' encoding='UTF-8'?> <xsl:stylesheet xmlns:xsl='http://
www.w3.org/1999/XSL/Transform' version='1.0'> <xsl:output method = 'text'
omit-xml-declaration='yes' indent='no'/> <xsl:template match='text()'> </
xsl:template>
```

Example

```
xsl-stylesheet-prolog = <?xml version='1.0' encoding='UTF-8'?>
<xsl:stylesheet xmlns:xsl='http://www.w3.org/1999/XSL/Transform'
version='1.0'> <xsl:output method = 'text' omit-xml-declaration='yes'
indent='no'/> <xsl:template match='text()'> </xsl:template>
```

[aznapi-decision-app] stanza

Use the [aznapi-decision-app] stanza to configure and enable the authorization REST API.

max-cache-size

Use the `max-cache-size` to set the maximum number of user credentials to be cached by the authorization decision application.

Syntax

```
max-cache-size = number_of_credentials_to_cache
```

Description

The maximum number of credentials which can be cached. If the addition of a new credential exceeds this maximum cache size a least-recently-used algorithm is used to remove an older entry, making room for the new credential.

Options

number_of_credentials_to_cache
An integer value which indicates the maximum number of user credentials to cache. The minimum value is 0. When the value is 0 no credentials will be cached.

Usage

This stanza entry is required.

Default value

8192

max-cache-lifetime

Use `max-cache-lifetime` to set the maximum length of time in seconds that a credential is cached by the authorization decision application.

Syntax

```
max-cache-lifetime = credential_lifetime_in_seconds
```

Description

The maximum lifetime, in seconds, for a credential which is stored in the credential cache.

Options

credential_lifetime_in_seconds
An integer value which indicates the maximum lifetime, in seconds, of a cached user credential. The minimum value is 0. When the value is 0 the credential is not cached.

Usage

This stanza entry is required.

Default Value

300

[azn-decision-info] stanza

Use the **[azn-decision-info]** stanza to define any extra information for the authorization framework to use when it is making authorization decisions. This extra information is obtained from elements of the HTTP request.

azn-decision-info

Use the **azn-decision-info** stanza entry to add extra information from the HTTP request, such as the method, to the authorization decision information.

Syntax

```
<attr-name> = <http-info>
```

Description

This stanza defines any extra information that is available to the authorization framework when it makes authorization decisions. This extra information can be obtained from the following elements of the HTTP request:

- HTTP method
- HTTP scheme
- HTTP cookies
- Request URI
- Client IP address
- HTTP headers
- Query string
- POST data

If the requested element is not in the HTTP request, no corresponding attribute is added to the authorization decision information.

Options

<attr-name>
The name of the attribute that contains the HTTP information.

<http-info>
The source of the information. It can be one of the following values:

- `method`
- `scheme`
- `uri`
- `client_ip`
- `header:`*<header-name>*

  Where *<header-name>* is the name of the header that contains information for WebSEAL to add to the authorization decision information. For example, `Host`.
- `cookie:`*<cookie-name>*

  Where*<cookie-name>* is the name of the cookie that contains information for WebSEAL to add to the authorization decision information.
- `query-arg:`*<query-arg-name>*

  Where *<query-arg-name>* is the query string parameter that contains information for WebSEAL to add to the authorization decision information. This entry indicates to WebSEAL that the query string parameters that are part of the requested resource URL are used for the access decision. If the specified value is found, it is sent to the EAS for the decision-making process.
- `post-data:`*<post-data-name>*

  Where the format of the *<post-data-name>* depends on the type of POST data.

  WebSEAL supports two types of POST data:

  – Normal FORM data, which is the `application/x-www-form-urlencoded` content-type.

    To add normal FORM data to the HTTP request, use the following format for this entry:

    ```
    post-data:<post-data-name>
    ```

    Where the *<post-data-name>* is the name of the selected form data field in the request. WebSEAL adds the corresponding value for this field to the authorization decision information.

– JavaScript Object Notation (JSON) data, which is the `application/json` content-type. For more information about the JSON syntax, see http://www.json.org.

To search for a key in the JSON data and add its value to the HTTP request, use the following format:

```
post-data:/"<JSON-node-id>"[[/"<JSON-node-id>"][<JSON-array-indx>]]...
```

Where:

"<JSON-node-id>"
The name of a node in the JSON data.

JSON data is essentially a hierarchy of name-value pairs. The forward slash character (/) that precedes each *"<JSON-node-id>"* identifies a level of the JSON hierarchy. You can repeatedly add *[/<JSON-node-id>]* elements to move through the JSON data hierarchy. Identify the node that contains the value that you want WebSEAL to add to the authorization decision information.

Each *<JSON-node-id>* must be:

- Enclosed in double quotation marks.
- Preceded by a forward slash character (/).
- A case-sensitive match with a node in the JSON data hierarchy.

  If WebSEAL does not find a matching node name in the POST data, no corresponding attribute is added to the authorization decision information.

<JSON-array-indx>

The contents of a node in the JSON data might be a JSON array. If you configure WebSEAL to search for a JSON node that contains an array, specify the array index of the value that you want WebSEAL to use. Use a base of 0. In other words, the first entry in the array has an index of `0`.

Note: The *<JSON-array-indx>* is not enclosed in double quotation marks.

Usage notes:

- The square brackets ([]) in this syntax indicate an optional element. Do not include square brackets in your configuration entry. Similarly, the ellipsis (...) indicates that you can repeat the optional elements that precede it. Do not include the ellipsis in your configuration entry.
- WebSEAL returns only node values of the following JSON types:
  - `String`
  - `Number`
  - `true` or `false`
  - `null`

  If the value of the selected node is not one of the types in this list, WebSEAL does not return it as authorization decision information.

  Object and Array types cannot be added to the authorization decision information.

Usage

This stanza entry is optional.

Default value

None.

Example 1: Standard HTTP elements

```
HTTP_REQUEST_METHOD = method
HTTP_HOST_HEADER= header:Host
```

If these example configuration entries are set in the **[azn-decision-info]** stanza, WebSEAL adds the following attributes to the authorization decision information:

HTTP_REQUEST_METHOD
Contains the HTTP method.

HTTP_HOST_HEADER
Contains the data from the Host header.

Example 2: JSON POST data

For this example, consider the following JSON form data:

```
{ "userid": "jdoe",
  "transactionValue": "146.67",
  "accountBalances": {
        "chequing": "4345.45",
        "savings": "12432.23",
        "creditLine": "19999.12"
    }
}
```

The following configuration entries in the **[azn-decision-info]** stanza extract information from this JSON form data.

```
USERID = post-data:/"userid"
SAVINGS = post-data:/"accountBalances"/"savings"
```

The first entry prompts WebSEAL to search for the JSON node called **"userid"**. In this example, the value that is associated with the **"userid"** node is `jdoe`. WebSEAL adds this value to the HTTP request in an attribute called **USERID**.

When WebSEAL processes the second entry, it searches for a top-level JSON node called **"accountBalances"**. Under the **"accountBalances"** hierarchy, WebSEAL locates the **"savings"** JSON node. In the example data, the value that is associated with this node is `12432.23`. WebSEAL adds this value to the HTTP request in an attribute called **SAVINGS**.

WebSEAL adds the following attributes to the authorization decision information:

USERID
Contains the value `jdoe`.

SAVINGS
Contains the value `12432.23`.

Example 3: JSON POST data with a JSON array value

For this example, consider the following JSON form data:

```
{
    "userid": "jdoe",
    "transactionValue": "146.67",
    "accounts": [
        {"name": "chequing", "balance": "4350.45"},
        {"name": "savings", "balance": "4350.46"}
    ]
}
```

The following configuration entry is included in the **[azn-decision-info]** stanza:

```
SAVINGSBAL = post-data:/"accounts"/1/"balance"
```

WebSEAL processes this entry as follows:

1. Searches for a top-level node in the JSON data called **"accounts"**.
2. Locates the element in position 1 of the JSON array (base 0).
3. Searches for the **"balance"** name-value pair in this array element.
4. Adds the associated value to the authorization decision information.

In this example, WebSEAL adds the following attribute to the authorization decision information:

SAVINGSBAL
Contains the value `4350.46`.

Example 4: Query string

For this example, consider the following entry in the **[azn-decision-info]** stanza:

```
urn:company:user:name = query-arg:username
```

If a user attempts to access the EAS protected resource:

```
http://www.example.com/t1.html?type=1&username=ann&dept=test
```

WebSEAL sends the following value to the EAS for use in the decision-making process:

```
urn:company:user:name=ann
```

[aznapi-external-authzn-services] stanza

Use the **[aznapi-external-authzn-services]** stanza to configure the OAuth External Authorization Service (EAS).

policy-trigger

Use the ***policy-trigger*** stanza entry to define the external authorization service.

Syntax

```
policy-trigger = plug-in_location [-weight N [& plug-in_parameters]]
```

Description

Defines the external authorization service.

Options

policy-trigger

Any string that is recognized as a valid key name. Stanza key names cannot contain white space or the open bracket ([) and close bracket (]) characters. The bracket characters are used to define new stanza names. The policy-trigger is case-sensitive for action set definitions because the actions themselves are case-sensitive. However, the policy-trigger is not case-sensitive if the trigger is a protected object policy (POP) attribute.

plug-in_location

The path name to the shared library or DLL module that contains the implementation of the plug-in for the specified policy trigger. The path name can be in a truncated form if the external authorization service is to be loaded by clients on multiple platforms. In this case, the service dispatcher searches for the plug-in using platform-specific prefixes and suffixes to match DLL names.

The name of the OAuth EAS plug-in is **amwoautheas**. For example:

```
libamwoautheas.so
```

N

The weight parameter is an unsigned **size_t** value and is optional. The value signifies the weight that any decision returned by this external authorization service is given in the entire decision process.

plug-in_parameters

Optionally, the external authorization service can be passed more initialization information in the form of arguments. The arguments must be preceded by the ampersand "&". The authorization service takes the remainder of the string that follows the ampersand &, breaks up the string into white space separated tokens, and passes the tokens directly to the administration service's initialization interface, `azn_svc_initialize()`, in the **argv** array parameter. The number of strings in the **argv** array is indicated by the **argc** function parameter.

A single parameter is required by the OAuth EAS. This parameter corresponds to the name of the OAuth EAS configuration file. That is, the file that contains the **[oauth-eas]** stanza and the corresponding **[tfim-cluster:<cluster>]** stanza.

Usage

This stanza entry is required when you are configuring OAuth EAS authentication.

Default value

None.

Example

The following example is an operation-based trigger with a user-defined action group of Printer and the actions rxT in that group. To specify the primary action group, specify only `:rxT`. The primary action group can be represented with an empty action group name or the string primary can be used explicitly. All lowercase letters are required if primary is used explicitly. Any policy-trigger that does not contain a colon (`:`) character is considered to be a POP attribute name.

```
Printer:rxT = eas_plugin -weight 60 & -server barney
```

The following example is for a POP attribute trigger called **webseal_pop_trigger**. When a POP that contains a reference to this string is encountered, the appropriate external authorization service is called to take part in the access decision.

```
webseal_pop_trigger = eas_plugin_2 -weight 70 & -hostname fred
```

Note: In order for the above POP attribute trigger to work, POP configuration must first be completed by the secure domain administrator, by using the **pdadmin pop** commands.

The following is an example configuration for the OAuth EAS, where the file `oauth_eas.conf` contains the **[oauth-eas]** stanza and the corresponding **[tfim-cluster:<cluster>]** stanza. This example is entered as one line in the WebSEAL configuration file:

```
webseal_pop_trigger = libamwoautheas.so & oauth_eas.conf
```

[ba] stanza

Use the **[ba]** stanza to configure basic authentication.

ba-auth

Use the **ba-auth** stanza entry to enable basic authentication.

Syntax

```
ba-auth = {none|http|https|both}
```

Description

Enables the use of the Basic Authentication mechanism for authentication.

Options

{none|http|https|both}

Specifies which protocols are supported. The value `both` means both HTTP and HTTPS.

Usage

This stanza entry is required.

Default value

`https`

Example

```
ba-auth = https
```

basic-auth-realm

Use the **basic-auth-realm** stanza entry to specify the basic authentication realm name.

Syntax

```
basic-auth-realm = realm_name
```

Description

String value that specifies the realm name.

Options

realm_name

This name is displayed in the browser dialog box when the user is prompted for login information. The string must consist of ASCII characters and can contain spaces.

Usage

This stanza entry is optional.

Default value

`IBM Security Verify Access for Web`

Example

```
basic-auth-realm = IBM Security Verify Access for Web
```

[certificate] stanza

Use the **[certificate]** stanza to configure certificate authentication.

accept-client-certs

Use the **accept-client-certs** stanza entry to control how WebSEAL handles client certificates from HTTPS clients.

Syntax

```
accept-client-certs = {never|critical|required|optional|prompt_as_needed}
```

Description

Specifies how to handle certificates from HTTPS clients.

Options

never
Never request a client certificate.

critical
Always request a client certificate. If a valid certificate is not presented, the SSL handshake fails.

required
Always request a client certificate. If a valid certificate is not presented, the SSL handshake succeeds but an error HTTP response is sent back to the client.

optional
Always request a client certificate. If a valid certificate is presented, use it.

prompt_as_needed
Only prompt for and process certificates when certificate authentication is necessary. An example of such situation is an ACL or POP check failure.

Note:

- When this value is set, ensure that the **ssl-id-sessions** stanza entry in the **[session]** stanza is set to no.
- The Alternate Port Method is required when Web Reverse Proxy is configured to accept HTTP/2 requests.
- The Alternative Port Method is required for TLSv1.3 clients.

Usage

This stanza entry is required.

Default value

never

Example

```
accept-client-certs = never
```

cert-cache-max-entries

Use the **cert-cache-max-entries** stanza entry to specify the maximum number of concurrent entries in the Certificate SSL ID cache.

Syntax

```
cert-cache-max-entries = number_of_entries
```

Description

Maximum number of concurrent entries in the Certificate SSL ID cache.

Options

number_of_entries
There is no absolute maximum size for the cache. However, the size of the cache cannot exceed the size of the SSL ID cache. A maximum size of 0 allows an unlimited cache size.

Usage

This stanza entry is required only when the **accept-client-certs** key is set to `prompt_as_needed`.

Default value

`1024`

Example

```
cert-cache-max-entries = 1024
```

cert-cache-timeout

Use the **cert-cache-timeout** stanza entry to specify the maximum lifetime, in seconds, for an entry in the Certificate SSL ID cache.

Syntax

```
cert-cache-timeout = number_of_seconds
```

Description

Maximum lifetime, in seconds, for an entry in the Certificate SSL ID cache.

Options

number_of_seconds
The minimum value is zero (`0`). A value of zero mean that when the cache is full, the entries are cleared based on a Least Recently Used algorithm.

Usage

This stanza entry is required only when the **accept-client-certs** key is set to `prompt_as_needed`.

Default value

`120`

Example

```
cert-cache-timeout = 120
```

cert-prompt-max-tries

Use the **cert-prompt-max-tries** stanza entry to specify how many times WebSEAL attempts to negotiate the SSL certificate before it assumes that the client cannot provide a certificate.

Syntax

```
cert-prompt-max-tries = number_of_tries
```

Description

During certificate authentication, WebSEAL prompts the browser to present the client's certificate. The SSL certificate negotiation process requires that the browser open and use a new (not existing) TCP connection.

Browsers typically maintain several open TCP connections to a given server. When WebSEAL tries to prompt the browser for a certificate, the browser often tries to reuse an existing TCP connection instead of opening a new TCP connection. Therefore, the prompting process must be retried. WebSEAL might need to prompt for a certificate several times before the browser opens a new TCP connection and allows the prompting process to succeed.

This configuration option controls how many times WebSEAL attempts to begin the SSL certificate negotiation process with the browser before assuming the client cannot provide a certificate.

Options

number_of_tries

Set the value to 5 because most browsers maintain a maximum of four TCP connections to a Web server. As each attempt by the browser to process the certificate prompts on an existing TCP connection fails, that TCP connection is closed. On the fifth attempt, with all TCP connections closed, the browser's only option is to open a new TCP connection.

If the value is set to less to 5, intermittent failures of certificate authentication might occur because the browser reuses existing TCP connections instead of opening a new TCP connection. These failures are more likely to occur in environments where login or other pages contain images that browsers access immediately before triggering the certificate prompts.

Values less than 2 or greater than 15 are not permitted.

This value is not used unless `accept-client-certs =prompt_as_needed`.

Usage

This stanza entry is required.

Default value

5

Example

```
cert-prompt-max-tries = 5
```

disable-cert-login-page

Use the **disable-cert-login-page** stanza entry to control whether WebSEAL bypasses the initial login page and directly prompts for the certificate.

Syntax

```
disable-cert-login-page = {yes|no}
```

Description

Determines whether the initial login page with an option to prompt for certificate is presented or if WebSEAL will bypass the page and directly prompt for the certificate.

Options

yes
The initial login page with an option to prompt for certificate is not presented; instead, WebSEAL bypasses this page and directly prompts for the certificate.

no
The initial login page with an option to prompt for certificate is presented.

Usage

This stanza entry is required.

Default value

no

Example

```
disable-cert-login-page = no
```

eai-data

Use the **eai-data** stanza entry to specify which client certificate data elements are passed to the external authentication interface (EAI) application by WebSEAL.

Syntax

```
eai-data = data:header_name
```

Description

The client certificate data elements that will be passed to the EAI application. Multiple pieces of client certificate data can be passed to the EAI application by including multiple eai-data configuration entries.

Options

header_name
Used to indicate the name of the HTTP header which will contain the data.

data
Used to indicate the data that will be included in the header. It should be one of the following:

- AlternativeDirectoryName

- AlternativeDNSName
- AlternativeIPAddress
- AlternativeURI
- AlternativeEmail
- Base64Certificate
- SerialNumber
- SubjectCN
- SubjectLocality
- SubjectState
- SubjectCountry
- SubjectOrganization
- SubjectOrganizationalUnit
- SubjectDN
- SubjectPostalCode
- SubjectEmail
- SubjectUniqueID
- IssuerCN
- IssuerLocality
- IssuerState
- IssuerCountry
- IssuerOrganization
- IssuerOrganizationUnit
- IssuerDN
- IssuerPostalCode
- IssuerEmail
- IssuerUniqueID
- Version
- SignatureAlgorithm
- ValidFrom
- ValidFromEx
- ValidTo
- ValidToEx
- PublicKeyAlgorithm
- PublicKey
- PublicKeySize
- FingerprintAlgorithm
- Fingerprint

Usage

This stanza entry is required for EAI based client certificate authentication.

Default value

no

Example

```
eai-data = SubjectCN:eai-cn
eai-data = SubjectDN:eai-dn
```

eai-uri

Use the **eai-uri** stanza entry to specify the URI of the external authentication interface (EAI) application that WebSEAL can use for certificate authentication. Configure this entry if you do not want to use the standard CDAS authentication mechanism.

Syntax

```
eai-uri = uri
```

Description

The resource identifier of the application that is called to perform the certificate authentication. This URI must be relative to the root web space of the WebSEAL server. If this configuration entry is not defined, the standard CDAS authentication mechanism is used to handle the authentication.

Options

uri

The resource identifier of the application that is called to perform the certificate authentication. This URI must be relative to the root web space of the WebSEAL server.

Note: A special value of *%lua-eai%* is used to indicate that a Lua transformation rule is to be used to perform the certificate authentication.

Usage

This stanza entry is required for EAI-based client certificate authentication.

Default value

no

Example

```
eai-uri = /jct/cgi-bin/eaitest/eaitest.pl
```

omit-root-cert

By default the complete certificate chain is sent as part of an SSL/TLS Certificate Message. An optional mode is allowed by the TLS RFC in which the root certificate (anchor) is omitted from the Certificate Message. Setting this option to `true` causes the root cert to be omitted from the message.

Syntax

```
omit-root-cert = {true|false}
```

Description

This configuration entry is a global option and affects all web connections to the server.

Options

`true`
Omit the root certificate from the message.

`false`
Do not omit the root certificate from the message.

Usage

This stanza entry is optional.

Default value

`false`

Example

```
omit-root-cert = false
```

[cert-map-authn] stanza

Use the **`[cert-map-authn]`** stanza to configure the Cross Domain Authentication Service (CDAS).

debug-level

Use the **`debug-level`** stanza entry to control the trace level for the authentication module.

Syntax

```
debug-level = level
```

Description

Controls the trace level for the authentication module.

Options

level
Specifies the initial trace level, with 1 designating a minimal amount of tracing and 9 designating the maximum amount of tracing.

Note: You can also use the Security Verify Access **pdadmin** trace commands to modify the trace level by using the trace component name of `pd.cas.certmap`. This trace component is only available after the first HTTP request is processed.

Usage

This stanza entry is optional.

Default value

`0`

Note: A debug level of `0` results in no tracing output.

Example

```
debug-level = 5
```

rules-file

Syntax

```
rules-file = file-name
```

Description

The name of the rules file that the CDAS can use for certificate mapping.

Options

file-name
The name of the rules file for the certificate mapping CDAS.

Usage

This stanza entry is required.

Default value

None.

Example

```
rules-file = cert-rules.txt
```

[cfg-db-cmd:entries] stanza

Use the **[cfg-db-cmd:entries]** stanza to define configuration entry settings when you use the **server sync** commands.

stanza::*entry*

Use the **stanza::entry** stanza entry to specify the configuration entries that are imported or exported from the configuration database.

Syntax

```
stanza::entry = {include|exclude}
```

Description

Each configuration entry is checked sequentially against each item in the `[cfg-db-cmd:entries]` stanza until a match is found. This first match then controls whether the configuration entry is included in, or excluded from, the configuration database. If no match is found, the configuration entry is excluded from the configuration database.

Syntax

entry
This field defines the stanza entry to be included or excluded. It may contain any pattern matching characters.

stanza
This field defines the stanza containing the data entry to be included or excluded. It may contain any pattern matching characters.

Options

include
Include the specified configuration entries when importing or exporting data from the configuration database.

exclude
Exclude the specified configuration entries when importing or exporting data from the configuration database.

Usage

This stanza entry is not required.

Default value

WebSEAL uses the values configured in the WebSEAL configuration file. See the WebSEAL configuration file template for the default entries.

Example

```
server::worker-threads = include
ldap::* = exclude
 *::* = include
```

[cfg-db-cmd:files] stanza

Use the **[cfg-db-cmd:files]** stanza entry to define settings for files that are included in the configuration database.

files

Use the **files** stanza entry to specify the files that WebSEAL includes in the configuration database. These files are replicated across WebSEAL servers in a cluster. Replication occurs when the **pdadmin server task cluster restart** command runs.

Syntax

Either:

```
files = cfg(stanza::entry)
```

Or:

```
files = file_name
```

Description

Defines the files that will be included (that is, imported or exported) in the configuration database.

Options

stanza
This field specifies the name of the stanza that contains the entry with the name of the file to be included in the configuration database. The configuration value defined by *stanza* and *entry* must contain the name of the file.

entry
This field specifies the stanza entry that contains the name of the file to be included in the configuration database. The configuration value defined by *stanza* and *entry* must contain the name of the file.

file_name
The name of the file.

Usage

This stanza entry is not required.

Default value

```
files = cfg(ssl::webseal-cert-keyfile)
files = cfg(ssl::webseal-cert-keyfile-stash)
files = cfg(junction::jmt-map)
files = cfg(server::dynurl-map)
```

Example

```
files = cert-rules.txt
files = jmt.conf
files = cfg(junction::jmt-map)
```

[cluster] stanza

Use the **[cluster]** stanza to configure a clustered WebSEAL server environment.

Notes:

- It is vital that this configuration stanza is not included in the configuration database. The `cluster::* = exclude` configuration entry in the **[cfg-db-cmd:entries]** stanza ensures this exclusion.
- In addition to the configuration entries listed here, a **config-version** entry is added at run time in a clustered environment. This configuration entry contains version information about the current configuration. Do NOT manually edit this version information.
- All cluster members must be the **same** server *type*. You can cluster either:
  - WebSEAL servers that are running on appliances.
  - WebSEAL servers that are running on standard operating systems.

is-master

Use the **is-master** stanza entry to set the master server for the WebSEAL cluster.

Syntax

```
is-master = {yes|no}
```

Description

Is this server the master for the WebSEAL cluster? You need to have a single master for each cluster. Any modifications to the configuration of a cluster must be made on the master.

Options

yes

This server is the master for the WebSEAL cluster.

no

This server is not the master for the WebSEAL cluster. The name of the master server must be specified in the **master-name** configuration entry that is also in the **[cluster]** stanza.

Usage

This stanza entry is required in a clustered environment. This stanza entry is not required for a single server environment.

Default value

There is no default value.

Example

```
is-master = no
```

master-name

Use the **master-name** stanza entry to define the authorization server name of the master for the WebSEAL cluster.

Syntax

```
master-name = azn-name
```

Description

Defines the authorization server name of the master for the WebSEAL cluster.

Options

azn-name

The authorization server name of the master.

Usage

This stanza entry is required if the value for **is-master** (also in the **[cluster]** stanza) is set to no. If the **is-master** entry is set to yes, WebSEAL ignores this **master-name** entry.

Default value

There is no default value.

Example

```
master-name = default-webseald-master.ibm.com
```

max-wait-time

Use the **max-wait-time** stanza entry to specify the maximum amount of time, in seconds, that the master server waits for a replica server to restart.

Syntax

```
max-wait-time = number
```

Description

Specifies the maximum amount of time to wait, in seconds, for a replica server to be restarted. This configuration entry is only applicable to the master server.

Options

number

The maximum number of seconds to wait for a replica server to be restarted.

Usage

This configuration entry is required if **is-master** (also in the **[cluster]** stanza) is set to **yes**.

Default value

60

Example

```
max-wait-time = 60
```

[compress-mime-types] stanza

Use the **[compress-mime-types]** stanza to control the HTTP data compression for specific MIME types.

mime_type

Syntax

```
mime_type = minimum_doc_size:[compression_level]
```

Description

Enables or disables HTTP compression based on the mime-type of the response and the size of the returned document.

Options

mime_type

The *mime_type* can contain a wild card pattern such as an asterisk (*) for the subtype, or it can be "*/*" to match all mime-types.

minimum_doc_size
The *minimum_doc_size* is an integer than can be positive, negative or zero. A size of -1 means do not compress this mime-type. A size of 0 means to compress the document regardless of its size. A size greater than 0 means to compress the document only when its initial size is greater than or equal to *minimum_doc_size*.

compression_level
The compression_level is an integer value between 1 and 9. The larger number results in a higher amount of compression. When compression-level is not specified, a default level of 1 is used.

Usage

This stanza entry is optional.

Default value

`*/* = -1`

Example

```
image/* = -1
text/html = 1000
```

[compress-user-agents] stanza

Use the **[compress-user-agents]** stanza to explicitly enable or disable compression for various browsers. You can configure WebSEAL to use the user-agent header that is sent by the client to determine whether to enable or disable HTTP compression.

pattern

Syntax

```
pattern = {yes|no}
```

Description

Enables or disables HTTP compression based on the user-agent header sent by clients. This entry is used to disable compression for clients which send an "accept-encoding: gzip" HTTP header but do not actually handle gzip content-encodings properly. An example of a user agent is a browser, such as Microsoft Internet Explorer 6.0

Options

yes
Enables HTTP compression based on the user-agent header sent by clients.

no
Disables HTTP compression based on the user-agent header sent by clients.

Usage

This stanza entry is optional.

Default value

None.

Example

```
*MSIE 6.0* = yes
```

[content] stanza

Use the **[content]** stanza to specify the format of macro data strings that are inserted into HTML server response pages. The data is inserted into the WebSEAL HTML files in either UTF-8 format or local code page format.

utf8-template-macros-enabled

Syntax

```
utf8-template-macros-enabled = {yes|no}
```

Description

Specifies how standard WebSEAL HTML files, such as login.html, have data inserted into them when %MACRO% strings are encountered.

This entry affects files in the `management` and `errors` directories. You can manage these directories from the Manage Reverse Proxy Management Root page of the LMI.

WebSEAL HTML pages use a UTF-8 character set by default. If you modify the character set to specify the local code page, set this entry to no.

Options

yes
When set to `yes`, data is inserted in UTF-8 format.

no
When set to no, data is inserted in the local code page format.

Usage

This stanza entry is required.

Default value

`yes`

Example

```
utf8-template-macros-enabled = yes
```

[content-cache] stanza

Use the **[content-cache]** stanza to configure WebSEAL content caching.

MIME_type

Syntax

```
MIME_type = cache_type:cache_size:maximum_age
```

Description

List of entries that define the caches which WebSEAL uses to store documents in memory.

Options

MIME_type
Any valid MIME type conveyed in an HTTP `Content-Type:` response header. This value may contain an asterisk to denote a wildcard (*). A value of */* represents a default object cache that holds any object that does not correspond to an explicitly configured cache.

cache_type
Defines the type of backing store to use for the cache. Only `memory` caches are supported.

cache_size
The maximum size, in kilobytes, to which the cache grows before objects are removed according to a least-recently-used algorithm. The minimum allowable value is 1 kilobyte. WebSEAL reports an error and fails to start if the value is less than or equal to zero (0). WebSEAL does not impose a maximum allowable value.

def-max-age
Specifies the maximum age (in seconds) if expiry information is missing from the original response. If no value is provided, a default maximum age of 3600 (one hour) will be applied. The configured default maximum age is only used when the cached response is missing the cache control headers: Cache-Control, Expires, and Last-Modified.

Note: If only Last-Modified is present, the maximum age will be calculated as ten percent of the difference between the current time and the last-modified time.

Usage

This stanza entry is optional.

Default value

None.

Example

```
text/html = memory:2000:3600
# image/* = memory:5000:3600
# */* = memory:1000:3600
```

[content-encodings] stanza

Use the **[content-encodings]** stanza to map document extensions to encoding types. WebSEAL uses this mapping to determine the correct MIME type to report in its response content-type header for local junction files.

extension

Use the *extension* stanza entry to map document extensions to encoding types. WebSEAL uses this mapping to report the correct MIME type in its response content-type header for local junction files.

Syntax

```
extension = encoding_type
```

Description

Entries in this stanza map a document extension to an encoding type. This mapping is used by WebSEAL to report the correct MIME type in its response content-type header for local junction files. This mapping is necessary so that WebSEAL can communicate to a browser that encoded (binary) data is being returned.

The MIME types defined in this stanza must also be defined in **[content-mime-types]**.

When WebSEAL encounters a document with two extensions, such as: `.txt.Z`, it produces two headers:

```
content-type: text/plain
content-encoding: x-compress
```

Thus even though the data is compressed, the response to the browser says *text/plain*. However, the extra content-encoding header tells the browser that the data is *compressed text/plain*.

In most cases, the administrator does not need to add additional entries. However, if the administrator introduces a new extension type that requires more than a text/plain response, the *extension* and *encoding_type* should be added to this stanza.

Options

encoding_type
Encoding type.

Usage

This stanza entry is required.

Default value

```
gz = x-gzip
Z = x-compress
```

Example

```
gz = x-gzip
Z = x-compress
```

[content-mime-types] stanza

Use the **[content-mime-types]** stanza to define the MIME type for specific document extensions.

deftype

Use the **deftype** stanza entry to specify the default type that WebSEAL assigns to pages that do not match any of the *extension = MIME_type* entries in the **content-mime-types** stanza.

Syntax

```
deftype = MIME_type
```

Description

Default type to assign to pages that do not match any of the *extension = MIME_type* entries defined in this stanza.

Options

MIME_type
Default type to assign to pages that don't match any of the *extension = MIME_type* entries defined in this stanza.

Usage

This stanza entry is required.

Default value

`text/plain`

The administrator should not change this value.

Example

```
deftype = text/plain
```

extension

Syntax

```
extension = MIME_type
```

Description

This stanza defines the MIME type for specific document extensions. The stanza contains a list of *extension = MIME_type* pairs. Many common MIME types are defined by default. Administrators can add additional entries. Both *extensions* and MIME_types must be declared using the ASCII character set. The entry of invalid MIME types does not affect WebSEAL, but may cause difficulty for client browsers.

Options

extension
The file name extension of documents of this MIME type.

MIME_type
The corresponding MIME type.

Usage

The entries in this stanza are required.

Default value

The following MIME types are defined by default:

```
html = text/html
htm = text/html
gif = image/gif
jpeg = image/jpeg
ps = application/postscript
shtml = text/x-server-parsed-html
jpg = image/jpeg
jpe = image/jpeg
mpeg = video/mpeg
mpe = video/mpeg
mpg = video/mpeg
bin = application/octet-stream
exe = application/octet-stream
Z = application/octet-stream
EXE = application/octet-stream
dll = application/octet-stream
DLL = application/octet-stream
ivsrv = application/octet-stream
pdf = application/pdf
au = audio/basic
snd = audio/basic
aiff = audio/x-aiff
aifc = audio/x-aiff
aif = audio/x-aiff
wav = audio/x-wav
ai = application/postscript
eps = application/postscript
rtf = application/rtf
zip = application/zip
```

```
ief = image/ief
tiff = image/tiff
tif = image/tiff
ras = image/x-cmu-raster
pnm = image/x-portable-anymap
pbm = image/x-portable-bitmap
pgm = image/x-portable-graymap
ppm = image/x-portable-pixmap
rgb = image/x-rgb
xbm = image/x-xbitmap
xpm = image/x-xpixmap
xwd = image/x-xwindowdump
txt = text/plain
rtx = text/richtext
tsv = text/tab-separated-values
etx = text/x-setext
qt = video/quicktime
mov = video/quicktime
avi = video/x-msvideo
movie = video/x-sgi-movie
js = application/x-javascript
ls = application/x-javascript
mocha = application/x-javascript
wrl = x-world/x-vrml
dir = application/x-director
dxr = application/x-director
dcr = application/x-director
crt = application/x-x509-ca-cert
tar = application/x-tar
```

Example

```
zip = application/zip
```

[cookie-attributes] stanza

Use the cookie-attributes stanza to define static attributes which are added to matched cookies after WebSEAL has finished processing the request (including HTTP transformation) and just before they are passed back to the client. The configured attributes will replace any corresponding attribute which already exists in the cookie.

cookie-name-pattern

Use the <cookie-name-pattern> stanza entry to define the static attributes which will be added to the cookies before they are passed back to the client.

Syntax

```
<cookie-name-pattern> = {[+|-<user-agent-group>]}<attr-1>{;<attr-2>;...}
```

Description

Entries in this stanza define static attributes which are added to matching cookies before they are passed back to the client. A blocklist (-) or a allowlist (+) group of user agents can be added to control which user agents the cookie attributes are applied for. The user agent group is defined in the [user-agent-groups] configuration stanza. Multiple configuration entries can be configured to manage the static attributes for different cookies. The configuration order of the entries is significant in that only the first matching configuration entry is applied to a cookie.

Options

<cookie-name-pattern>
The <cookie-name-pattern> is used to match a cookie. The '*?' pattern matching characters can be used.

{[+|-<user-agent-group]}<attr-1>{;<attr-2>;...}
The attributes which are to be added to the cookie. A blocklist (-) or a allowlist (+) group of user agents can be added to control which user agents the cookie attributes are applied for. The user agent group is defined in the [user-agent-groups] configuration stanza. Multiple attributes can be specified, delimited by the ';' character.The supported attributes include: Comment, Expires, Max-Age, Domain, Path, Secure, HttpOnly, and SameSite.

Usage

This stanza entry is optional.

Default Value

None

Example

To add the 'SameSite=None' attribute to all cookies which are passed back to the client, but only for those clients not in the '[unsupported-same-site]' group:

```
* = [-unsupported-same-site]SameSite=None
```

[cors-policy:policy-name] stanza

Use this stanza to house configuration that is specific to a particular CORS policy. The <policy-name> component of the stanza name must be changed to the name the policy will be given.

request-match

Use this entry to define the pattern to be matched against the HTTP request line, which includes method, URI, and protocol.

Syntax

```
request-match = <request-line>
```

Description

This entry defines the pattern to be matched against the HTTP request line, which includes method, URI, and protocol.

You can also match a request by using a host header. Use this option to selectively enable this function for a particular virtual host junction. To selectively match an entry based on a particular host header, add a prefix to the `<request-line>` with the string `[<host>]`.

The CORS policy described in this stanza will be applied to any cross-origin or pre-flight requests which match these entries.

Options

request-line
Contains the request line to be matched against. The pattern matching is case-sensitive. You can use wildcard characters * and ?.

Usage

This stanza entry is required.

You can specify multiple entries if needed.

Default Value

None.

Example

```
request-match = GET /index.html HTTP/1.1
request-match = GET /jct/* *
request-match = [api.ibm.com]GET /login/*
```

allow-origin

The allow-origin entry specifies which origins presented by clients are permitted to make cross-origin requests to resources which this policy is applicable to.

Syntax

```
allow-origin = <origin>
```

Description

An origin which is permitted for this policy. This configuration entry may be specified multiple times to indicate multiple allowable origins. A value of '*' can be specified to indicate that requests are allowed from any origin.

When configured with an origin or list of origins, this configuration entry adds the following header to pre-flight requests:

```
Access-Control-Allow-Origin = <any value which matches the request's origin header>
```

When configured with '*', this configuration entry adds the following header to pre-flight requests:

```
Access-Control-Allow-Origin = <request's origin header>
```

This entry affects both pre-flight and cross-origin requests. This entry is used when validating cross-origin requests.

Options

origin
: This entry should either be '*' or a complete origin including a scheme, hostname and optionally any port information.

Usage

This stanza entry is required.

You can specify multiple entries if needed.

The origin matching performed is case sensitive.

If an '*' entry is specified, all other allow-origin entries for this policy is ignored.

Default Value

None.

Example

```
allow-origin = https://isva.ibm.com
allow-origin = https://isva.ibm.com:9443
or
allow-origin = *
```

allow-credentials

The allow-credentials entry controls whether or not the reverse proxy returns the Access-Control-Allow-Credentials header to clients.

Syntax

```
allow-credentials = {true, false}
```

Description

Indicates to clients whether authentication is required when accessing resources which are protected by this policy. This will indicate that the policy should insert the following header in both pre-flight and cross-origin responses:

```
Access-Control-Allow-Credentials = true
```

Note:

- Setting this entry to false or not specifying it omits the header from responses. The `Access-Control-Allow-Credentials` header is never present with any value other than true.

- If this entry is enabled and all origins are allowed (allow-origin is set to '*') the reverse proxy never responds with a wildcard for allowed origins:

  ```
  Access-Control-Allow-Origin: '*'
  ```

  When all origins are allowed and credentials are required, the reverse proxy will instead respond with the origin presented in the request as the allowed origin:

  ```
  Access-Control-Allow-Origin: <origin header from request>
  ```

This applies to both pre-flight and cross-origin requests.

Options

yes | true
Add the *Access-Control-Allow-Credentials, header with a value of true to pre-flight and cross origin requests.*

no | false
Do not add an *Access-Control-Allow-Credentials header to pre-flight and cross origin requests*.

Usage

This stanza entry is optional.

Default value

false

Example

```
allow-credentials = false
```

expose-header

The expose-header entry specifies which headers the reverse proxy returns, if any, in the `Access-Control-Expose-Headers` header to cross-origin requests.

Syntax

```
expose-header = <header-name>
```

Description

Response headers which clients are allowed to expose. This configuration entry might be specified multiple times to specify multiple headers that can be exposed. This adds the following header to responses to cross-origin requests:

```
Access-Control-Expose-Headers = <header1>, <header2>, ..., <headerN>
```

Note:

- All specified headers are returned as a comma separated list in a single Access-Control-Expose-Headers header.
- This value is not considered when processing cross-origin requests and this header is only used to advise the client of additional headers it may expose.

Options

header-name
This name of the header which should be added to the Access-Control-Expose-Header header.

Usage

This stanza entry is optional.

Default value

None

Example

```
expose-header = X-IBM-APP-NAME
expose-header = X-IBM-APP-VERSION
```

handle-pre-flight

Use this entry to control whether or not the reverse proxy generates responses to pre-flight requests.

Syntax

```
handle-pre-flight = {true, false}
```

Description

This entry defines whether or not the reverse proxy handles CORS pre-flight responses.

The reverse proxy is capable of generating pre-flight responses. The reverse proxy considers requests that meet all of the following criteria to be CORS pre-flight requests:

- the request matches this policy stanza. See “request-match” on page 66
- the request uses the *OPTIONS* method
- the request contains an *Origin* header specifying a permitted origin. See “allow-origin” on page 66
- the request contains an *Access-Control-Request-Method* header

The pre-flight response mechanism uses information from the following entries within a `[cors-policy:<policy-name>]` stanza to generate a response:

- allow-credentials
- allow-header
- allow-method
- max-age

Options

yes|true
The reverse proxy generates responses to requests it identifies as pre-flight cross-origin requests.

no|false
The reverse proxy will not respond to pre-flight cross-origin requests and forwards them to the backend application.

Usage

This stanza entry is optional.

Default Value

false

Example

```
handle-pre-flight = true
allow-credentials = true
allow-header = X-IBM-HEADER
max-age = 3600
```

allow-header

The allow-header entry specifies which headers are presented in preflight responses to clients as acceptable to use when making cross-origin requests to resources which this policy is applicable to.

Syntax

```
allow-header = <origin>
```

Description

A header which is permitted when making cross-origin requests to the resources protected by this policy. This configuration entry may be specified multiple times to indicate multiple allowable headers. A value of '*' can be specified to indicate that any header is acceptable, in this case the reverse proxy responds with any header(s) presented by the client in the Access-Control-Request-Headers header.

When configured with a header or list of headers, this configuration entry adds the following header to pre-flight responses:

```
Access-Control-Allow-Headers = <any values given in this policy>
```

When configured with '*', this configuration entry adds the following header to pre-flight responses:

```
Access-Control-Allow-Header = <any value presented by the client in the Access-Control-Request-
Headers>
```

Note: The following simple headers are always considered allowed and are never returned in the Access-Control-Allow-Headers header:

```
Accept
Accept-Language
Content-Language
Content-Type: application/x-www-form-urlencoded
Content-Type: multipart/form-data
Content-Type: text/plain
```

This entry only affects only pre-flight requests.

Options

header
This entry should either be '*' or a header name.

Usage

You can specify multiple entries if needed.

The header matching performed is not case sensitive.

If an '*' entry is specified, all other allow-header entries for this policy are ignored.

Default Value

None.

Example

```
allow-header = X-IBM-VERSION
allow-header = X-IBM-ROUTE
```

allow-method

The allow-method entry specifies which methods will be presented in preflight responses to clients as acceptable to use when making cross-origin requests to resources which this policy is applicable to.

Syntax

```
allow-method = <method>
```

Description

A method which is permitted when making cross-origin requests to the resources protected by this policy. This configuration entry may be specified multiple times to indicate multiple allowable methods. A value of '*' can be specified to indicate that any method is acceptable, in this case the reverse proxy will respond with the method presented by the client in the Access-Control-Request-Method header.

When configured with a header or list of headers, this configuration entry will add the following header to pre-flight responses:

```
Access-Control-Allow-Method = <any values given in this policy>
```

When configured with '*', this configuration entry will add the following header to pre-flight responses:

```
Access-Control-Allow-Method = <any value presented by the client in the Acccess-Control-Request-
Method>
```

Note:

- If a pre-flight request is received containing an Access-Control-Request-Method header with a value that is not acceptable, the reverse proxy will return a CORS error response.
- The following simple methods are considered always allowed and are never returned in the Access-Control-Allow-Method header:
  - *GET*
  - *HEAD*
  - *POST*

This entry only affects only pre-flight requests.

Options

header
This entry should either be '*' or a header name.

Usage

You can specify multiple entries if needed.

The method matching performed is case sensitive.

If an '*' entry is specified, all other allow-header entries for this policy are ignored.

Default Value

None.

Example

```
allow-method = PUT
```

max-age

The max-age entry specifies the amount of time that clients may cache the results of a pre-flight response.

Syntax

```
max-age = <seconds>
```

Description

The number of seconds a client should cache the results of a pre-flight check. This adds the following header to pre-flight responses:

```
Access-Control-Max-Age: <seconds>
```

This entry should be a positive integer or the values -1 or 0 which are special cases explained below.

- Setting this value to -1 will indicate to the client that it should not cache pre-flight results.
- Setting this value to 0 will allow the client to cache results at its own discretion.

If this entry is not present, no Access-Control-Max-Age header is added.

This entry only affects only pre-flight requests.

Options

seconds
The number of seconds a client should cache the results of a pre-flight check.

Usage

This stanza entry is optional.

Default Value

None.

Example

```
max-age = 3600
```

[cred-viewer-app] stanza

enable-embedded-html

Use the `enable-embedded-html` to enable or disable the generation of the credential viewer HTML response.

Syntax

```
enable-embedded-html = {yes|no}
```

Description

Whether the credential viewer application returns the HTML response page to render the credential data for requests which do not have the '`Accept: application/json`' header set.

Options

yes
The embedded credential viewer returns a HTML response for requests which do not have the '`Accept: application/json`' header.

no
The embedded credential viewer returns a 400 response for requests which do not have the '`Accept: application/json`' header.

Usage

This stanza entry is optional.

Default Value

yes

attribute-rule

The `attribute-rule` configuration entry specifies the rules which are used to determine which credential attributes should be included in the response.

Syntax

```
attribute-rule ={+|-}<attribute>
```

Description

The rules which define the credential attributes which will be included in the credential viewer response. This entry may be repeated multiple times, once for each rule which is to be defined.

Each attribute in the credential will be matched against each rule in order until a match is found. The corresponding prefix (+|-) will then be used to control whether the attribute is included or excluded from the response. If no matching rule is found the attribute will be included in the response.

The configuration entry could alternatively contain the name of a single credential attribute whose value is used to define the attribute rules. In this scenario each individual rule in the attribute should be separated by a space character. If only a single attribute-rule configuration entry is defined, and the entry does not start with a '+' or '-' character, it will be used as the name of the credential attribute which contains the attribute rules.

Options

+
Indicates that the attribute should be included.

-
Indicates that the attribute should be excluded.

<attribute>
The name of the attribute to which this rule applies (the '*?' pattern matching characters can be used), or the name of the attribute whose value contains the attribute rules.

Usage

This stanza entry is optional.

Default Value

None

Example

```
attribute-rule = -AUTHENTICATION_LEVEL
or
attribute-rule = FILTER_RULE
```

[credential-policy-attributes] stanza

Use the **[credential-policy-attributes]** stanza to specify the Security Verify Access policy values that are stored in credentials during authentication.

policy-name

Syntax

```
policy-name = credential-attribute-name
```

Description

Controls which Security Verify Access policy values are stored in credentials during authentication

Options

credential-attribute-name
Credential attribute name.

Usage

This stanza entry is optional.

Default value

None.

Example

```
AZN_POLICY_MAX_FAILED_LOGIN = tagvalue_max_failed_login
```

[credential-refresh-attributes] stanza

Use the **[credential-refresh-attributes]** stanza to configure the credential refresh behavior in WebSEAL.

attribute_name_pattern

Use the ***attribute_name_pattern*** stanza entry to specify whether attributes are preserved or refreshed during a credential refresh.

Syntax

```
attribute_name_pattern = {preserve|refresh}
```

Description

Specifies whether an attribute, or group of attributes that match a pattern, are preserved or refreshed during a credential refresh.

Options

preserve
Original attribute value is preserved in the new credential.

refresh
Original attribute value is refreshed in the new credential.

Usage

This stanza entry is optional.

Default value

`preserve`

Example

```
tagvalue_* = preserve
```

authentication_level

Use the **authentication_level** stanza entry to control whether the authentication level is preserved or refreshed during a credential refresh.

Syntax

```
authentication_level = {preserve|refresh}
```

Description

Specifies whether the authentication level for the user is preserved or refreshed during a credential refresh. The authentication level can reflect the results of an authentication strength policy (step-up authentication). In most cases, it is best to preserve this level during a credential refresh.

Options

preserve
The original attribute value is preserved in the new credential.

refresh
The original attribute value is refreshed in the new credential.

Usage

This stanza entry is required.

Default value

preserve

Example

```
authentication_level = preserve
```

[dsess] stanza

Use the **[dsess]** stanza to configure the distributed session cache.

dsess-sess-id-pool-size

Use the **dsess-sess-id-pool-size** stanza entry to set the maximum number of pre-allocated session IDs in the replica set.

Syntax

```
dsess-sess-id-pool-size = number
```

Description

The maximum number of session IDs that are pre-allocated within the replica set.

Note: This option is used by the **[dsess-cluster]** stanza.

Options

number
The maximum number of session IDs that are pre-allocated within the replica set.

Usage

This stanza entry is required when:

```
[session]
dsess-enabled = yes
```

Default value

125

Example

```
dsess-sess-id-pool-size = 125
```

dsess-cluster-name

Use the **dsess-cluster-name** stanza entry to specify the name of the distributed session cache cluster to which this server belongs.

Syntax

```
dsess-cluster-name = cluster name
```

Description

Specifies the name of the distributed session cache cluster to which this server belongs.

Options

cluster name
The name of the distributed session cache cluster to which this server belongs. This field must be defined and reference an existing **dsess-cluster** stanza qualified by the value of this entry.

Usage

This stanza entry is required when:

```
[session]
dsess-enabled = yes
```

Default value

dsess

Example

```
dsess-cluster-name = dsess
```

[dsess-cluster] stanza

Use the **[dsess-cluster]** to configure a distributed session cache cluster.

basic-auth-user

Use the **basic-auth-user** stanza entry to specify the user name that WebSEAL includes in the basic authentication header.

Syntax

```
basic-auth-user = user_name
```

Description

Specifies the name of the user that is included in the basic authentication header.

Options

user_name
The user name to include in the basic authentication header.

Usage

This stanza entry is optional.

Default value

None.

Example

```
basic-auth-user = admin
```

basic-auth-passwd

Use the **basic-auth-passwd** stanza entry to specify the password that WebSEAL includes in the basic authentication header.

Syntax

```
basic-auth-passwd = password
```

Description

Specifies the password for WebSEAL to include in the basic authentication header when it is communicating with the Federation Runtime.

Options

password
The password to include in the basic authentication header.

Usage

This stanza entry is optional.

Default value

None.

Example

```
basic-auth-passwd = myPassword1
```

gsk-attr-name

Use the **gsk-attr-name** stanza entry to specify additional GSKit attributes to use when initializing an SSL connection with the distributed session cache.

Syntax

```
gsk-attr-name = {enum | string | number}:id:value
```

Description

This configuration entry can be specified multiple times. Configure a separate entry for each GSKit attribute.

Options

{enum | string | number}
The GSKit attribute type.

id
The identity associated with the GSKit attribute.

value
The value for the GSKit attribute.

Usage

This stanza entry is optional.

See "Appendix: Supported GSKit attributes" on page 583 for a list of GSKit attributes that can be configured.

You cannot configure the following restricted GSKit attributes:

```
GSK_KEYRING_FILE,
GSK_KEYRING_PW,
GSK_KEYRING_STASH_FILE,
GSK_V2_SIDCACHE_SIZE,
GSK_V3_SIDCACHE_SIZE,
GSK_V2_SESSION_TIMEOUT,
GSK_V3_SESSION_TIMEOUT,
GSK_PROTOCOL_SSLV2,
GSK_PROTOCOL_SSLV3,
GSK_PROTOCOL_TLSV1,
GSK_PROTOCOL_TLSV11,
GSK_PROTOCOL_TLSV12,
GSK_LDAP_SERVER,
GSK_LDAP_SERVER_PORT,
GSK_LDAP_USER,
GSK_LDAP_USER_PW,
GSK_CRL_CACHE_SIZE,
GSK_CRL_CACHE_ENTRY_LIFETIME,
GSK_ACCELERATOR_NCIPHER_NF,
GSK_ACCELERATOR_RAINBOW_CS,
GSK_PKCS11_DRIVER_PATH,
GSK_PKCS11_TOKEN_LABEL,
GSK_PKCS11_TOKEN_PWD,
GSK_PKCS11_ACCELERATOR_MODE,
GSK_BASE_CRYPTO_LIBRARY,
GSK_OCSP_ENABLE,
GSK_OCSP_URL,
GSK_OCSP_NONCE_GENERATION_ENABLE,
GSK_OCSP_NONCE_CHECK_ENABLE,
GSK_OCSP_REQUEST_SIGKEYLABEL,
GSK_OCSP_REQUEST_SIGALG,
GSK_OCSP_PROXY_SERVER_NAME,
GSK_OCSP_PROXY_SERVER_PORT,
GSK_OCSP_RETRIEVE_VIA_GET,
GSK_OCSP_MAX_RESPONSE_SIZE,
GSK_USE_IO_EVENTS,
GSK_USER_DATA,
GSK_NO_RENEGOTIATION,
GSK_ALLOW_ANY_RENEGOTIATION,
GSK_ALLOW_ABBREVIATED_RENEGOTIATION,
GSK_ALLOW_ONLY_EXTENDED_RENEGOTIATION,
```

If you attempt to modify any of these attributes then an error message will be generated.

Default value

None.

Example

The following entry is for the GSKit attribute GSK_HTTP_PROXY_SERVER_NAME, which has an identity value of 225:

```
gsk-attr-name = string:225:proxy.ibm.com
```

See also

“gsk-attr-name” on page 508
“jct-gsk-attr-name” on page 511
“gsk-attr-name” on page 561

handle-idle-timeout

Syntax

```
handle-idle-timeout = number
```

Description

Limits the length of time that a handle remains idle before it is removed from the handle pool cache.

Options

number
The length of time, in seconds, before an idle handle will be removed from the handle pool cache.

Usage

This stanza entry is required when:

```
[session]
dsess-enabled = yes
```

Default value

240

Example

```
handle-idle-timeout = 240
```

handle-pool-size

Syntax

```
handle-pool-size = number
```

Description

The maximum number of idle Simple Access Object Protocol (SOAP) handles that the dsess client will maintain at any given time.

Options

number
The maximum number of idle SOAP handles that the dsess client will maintain at any given time.

Usage

This stanza entry is required when:

```
[session]
dsess-enabled = yes
```

Default value

10

Example

```
handle-pool-size = 10
```

load-balance

Controls the behavior when multiple servers with the same configured priority are available.

Syntax

```
load-balance = {yes | no}
```

Description

WebSEAL uses the same SOAP client for all outgoing SOAP communication. The SOAP client supports load balancing, so it is a valid configuration in the `[*-cluster]` stanzas.

However the DSC does not support load balancing. So while it can be configured on the client side, it does not function in `[dsess-cluster]`. Do not use load balancing in the `[dsess-cluster]` stanza.

Options

yes
All requests are sent to matching servers in a round-robin fashion.

no
All requests are sent to the first matching server that is available. The matching order is the order they appear in the configuration file.

Usage

This stanza entry is optional.

Default value

The default value is `yes`.

Example

```
load-balance = yes
```

max-wait-time

Use this entry to control the maximum length of time, in seconds, that the request will block while waiting for a server to become available.

Syntax

```
max-wait-time = <number>
```

Description

Specifies the maximum length of time, in seconds, that the request will block while waiting for a server to become available. This configuration entry can be used to help eliminate errors being returned to the client during a server failover.

Options

<number>
Length of time, in seconds, that the request will block while waiting for a server to become available.

Usage

This stanza entry is optional.

Default value

`0`

Example

```
max-wait-time = 0
```

response-by

Use the **response-by** stanza entry to specify the length of time in seconds that the dsess client waits for updates from the distributed session cache.

Syntax

```
response-by = seconds
```

Description

The length of time (in seconds) that the dsess client blocks to wait for updates from the distributed session cache.

Options

seconds
The length of time (in seconds) that the dsess client blocks to wait for updates from the distributed session cache.

Usage

This stanza entry is required when:

```
[session]
dsess-enabled = yes
```

Default value

60

Example

```
response-by = 60
```

server

Use the **server** entry in the **[dsess-cluster]** stanza to specify each distributed session cache server and its priority in the cluster.

Syntax

```
server = {[0-9],}<URL>
```

Description

Specifies a priority level and URL for each distributed session cache server that is a member of this cluster. Multiple server entries can be specified for a given cluster.

Options

0-9
A digit, 0-9, that represents the priority of the server within the cluster (9 being the highest, 0 being the lowest). If the priority is not specified, a priority of 9 is assumed.

Note: There can be no space between the comma (,) and the URL. If no priority is specified, the comma is omitted.

URL
A well-formed HTTP or HTTPS uniform resource locator for the server.

Usage

This stanza entry is required when:

```
[session]
dsess-enabled = yes
```

Default value

This entry is disabled by default.

Example

```
server = 9,http://dsc.example.com/DSess/services/DSess
```

ssl-fips-enabled

Use the **ssl-fips-enabled** entry in the **[dsess-cluster]** stanza to control whether WebSEAL uses TLSv1 or SSLv3 communication with the distributed session cache.

Syntax

```
ssl-fips-enabled = {yes|no}
```

Description

Determines whether Federal Information Process Standards (FIPS) mode is enabled on the distributed session cache. If no configuration entry is present, the setting from the global setting—as determined by the **ssl-fips-enabled** entry in the **[ssl]** stanza of the policy server—takes effect.

When set to yes or the setting in the policy server configuration file is set to yes, Transport Layer Security (TLS) version 1 (TLSv1) is the secure communication protocol used. When set to no or the setting in the policy server configuration file is set to no, SSL version 3 (SSLv3) is the secure communication protocol used.

Note: The **[dsess-cluster] ssl-nist-compliance** setting can override this entry. If **ssl-nist-compliance** is set to yes, FIPS mode processing is automatically enabled.

Options

yes
Indicates that TLSv1 is the secure communication protocol.

no
Indicates that SSLv3 is the secure communication protocol.

Usage

This stanza entry is optional.

Default value

None.

If a different FIPS level than that of the policy server is required, it is the responsibility of the administrator to edit the configuration file, uncomment the stanza entry, and specify this value.

Example

```
ssl-fips-enabled = yes
```

ssl-keyfile

Syntax

```
ssl-keyfile = file_name
```

Description

The name of the key database file, which houses the client certificate to be used.

Options

file_name
The name of the key database file that houses the client certificate for WebSEAL to use.

Usage

This stanza entry is only required if one or more of the cluster server URLs specified in the server entries uses SSL (that is, contains an HTTPS protocol specification in the URL). If no cluster server uses the HTTPS protocol, this entry is not required.

```
[session]
dsess-enabled = yes
```

Default value

None.

Example

```
ssl-keyfile = keyfile.kdb
```

ssl-keyfile-label

Syntax

```
ssl-keyfile-label = label_name
```

Description

The label of the client certificate within the key database.

Options

label_name
Client certificate label name.

Usage

This stanza entry is required when:

```
[session]
dsess-enabled = yes
```

Default value

None.

Example

```
ssl-keyfile-label = WS6
```

ssl-keyfile-stash

Syntax

```
ssl-keyfile-stash = file_name
```

Description

The name of the password stash file for the key database file.

Options

file_name
The password stash file.

Usage

This stanza entry is required when:

```
[session]
dsess-enabled = yes
```

Default value

None.

Example

```
ssl-keyfile-stash = pdsrv.sth
```

ssl-nist-compliance

Use the **ssl-nist-compliance** stanza entry in the **[dsess-cluster]** stanza to enable or disable NIST SP800-131A compliance for the distributed session cache.

Syntax

```
ssl-nist-compliance = {yes|no}
```

Description

Enables or disables NIST SP800-131A compliance for the distributed session cache.

Enabling NIST SP800-131A compliance results in the following automatic configuration:

- Enables FIPS mode processing.

  Note: When NIST SP800-131A compliance is enabled, FIPS mode processing is enabled regardless of the setting for the **[dsess-cluster] ssl-fips-enabled** configuration entry.
- Enables TLS v1.2.

  Note: TLS v1 and TLS v1.1 are not disabled.
- Enables the appropriate signature algorithms.
- Sets the minimum RSA key size to 2048 bytes.

If this **ssl-nist-compliance** configuration entry is not present, WebSEAL uses the global **nist-compliance** setting in the **[ssl]** stanza.

Options

yes
A value of yes enables NIST SP800-131A compliance.

no
A value of no disables NIST SP800-131A compliance.

Usage

This stanza entry is optional.

Default value

no

Example

```
ssl-nist-compliance = no
```

ssl-valid-server-dn

Syntax

```
ssl-valid-server-dn = certificate_DN
```

Description

Specifies the DN of the server (obtained from the server SSL certificate) that is accepted. If no entry is configured, any valid certificate signed by a CA in the key file is accepted.

Options

value
Specifies the DN of the server (obtained from the server SSL certificate) that is accepted. If no entry is configured, any valid certificate signed by a CA in the key file is accepted.

Usage

This stanza entry is required when:

```
[session]
dsess-enabled = yes
```

Default value

None.

Example

```
ssl-valid-server-dn = CN=Verify Access,OU=SecureWay,O=Tivoli,C=US
```

timeout

Use the **timeout** entry in the **[dsess-cluster]** stanza to specify the number of seconds that WebSEAL waits for a response from the distributed session cache.

Syntax

```
timeout = seconds
```

Description

The length of time, in seconds, to wait for a response to be received back from the distributed session cache.

Options

seconds
The length of time, in seconds, to wait for a response to be received back from the distributed session cache.

Usage

This stanza entry is required when:

```
[session]
dsess-enabled = yes
```

Default value

30

Example

```
timeout = 30
```

[eai] stanza

Use the **[eai]** stanza to configure the external authentication interface (EAI).

eai-auth

Use the **eai-auth** stanza entry to enable the external authentication interface.

Syntax

```
eai-auth = {none|http|https|both}
```

Description

Enables the external authentication interface.

Options

{none|http|https|both}
Enables the external authentication interface. No other external authentication interface parameters will take effect if set to "none".

Usage

This stanza entry is required.

Default value

none

Example

```
eai-auth = none
```

eai-auth-level-header

Use the **eai-auth-level-header** stanza entry to specify the name of the header that contains the authentication strength level for the generated credential.

Syntax

```
eai-auth-level-header = header-name
```

Description

Specifies the name of the header that contains the authentication strength level for the generated credential.

Options

header-name
The name of the header that contains the authentication strength level for the generated credential.

Usage

This stanza entry is optional.

Default value

am-eai-auth-level

Example

```
eai-auth-level-header = am-eai-auth-level
```

eai-create-multi-valued-attributes

Use this entry to specify whether multiple extended attribute headers of the same name are added to the credential as a multi-valued attribute or a single comma-delimited attribute.

Syntax

```
eai-create-multi-valued-attributes = {yes|no}
```

Description

This configuration entry is used to determine whether multiple extended attribute headers of the same name are added to the credential as a multi-valued attribute or a single comma-delimited attribute.

Options

yes
Add multiple extended attribute headers of the same name as a multi-valued attribute.

no
Add multiple extended attribute headers of the same name as a single comma-delimited attribute.

Usage

This stanza entry is optional.

Default value

no

Example

```
eai-create-multi-valued-attributes = no
```

eai-error-text-header

Use the `eai-error-text-header` stanza entry to specify the name of the header that contains the error text to be included in WebSEAL generated error responses.

Syntax

```
eai-error-text-header = header-name
```

Description

Specifies the name of the header that contains the error text to be included in WebSEAL generated error responses using the %ERR0R_TEXT% macro.

Options

header-name
The name of the header that contains the error text.

Usage

This stanza entry is optional.

Default Value

`am-eai-error-text`

Example

```
eai-error-text-header = am-eai-error-text
```

eai-ext-user-id-header

Use the **eai-ext-user-id-header** stanza entry to specify the name of the header that contains the ID of the external user to use when creating a credential.

Syntax

```
eai-ext-user-id-header = header-name
```

Description

Specifies the name of the header that contains the ID of the external (not in the Security Verify Access user registry) user to use when creating a credential.

Options

header-name
The name of the header that contains the ID of the external user that is used when generating a credential.

Usage

This stanza entry is optional.

Default value

`am-eai-ext-user-id`

Example

```
eai-ext-user-id-header = am-eai-ext-user-id
```

eai-ext-user-groups-header

Use the **eai-ext-user-groups-header** stanza entry to specify the name of the header that contains the group or groups an external user is to be considered a member of when generating a credential.

Syntax

```
eai-ext-user-groups-header = header-name
```

Description

Specifies the name of the header that contains the group or groups an external user is to be considered a member of when generating a credential. This entry is only used when the **eai-ext-user-id-header** stanza entry's value is provided.

Options

header-name
The name of the header that contains the group or comma separated list of groups, that the external user is to be considered a member of, when generating a credential.

Usage

This stanza entry is optional.

Default value

`am-eai-ext-user-groups`

Example

```
eai-ext-user-groups-header = am-eai-ext-user-groups
```

eai-pac-header

Use the **eai-pac-header** stanza entry to specify the name of the HTTP header that contains the authentication data in Privilege Attribute Certificate (PAC) format.

Syntax

```
eai-pac-header = header-name
```

Description

Specifies the name of the HTTP header that contains the authentication data that is returned from the external authentication interface server in PAC format.

Options

header-name
The name of HTTP header that contains the authentication data that is returned from the external authentication interface server in PAC format.

Usage

This stanza entry is optional.

Default value

`am-eai-pac`

Example

```
eai-pac-header = am-eai-pac
```

eai-pac-svc-header

Use the **eai-pac-svc-header** stanza entry to specify the name of the header that contains the service ID, which WebSEAL uses to convert the Privilege Attribute Certificate (PAC) into a credential.

Syntax

```
eai-pac-svc-header = header-name
```

Description

Specifies the name of the header that contains the service ID that is used to convert the PAC into a credential.

Options

header-name
The name of the header that contains the service ID that is used to convert the PAC into a credential.

Usage

This stanza entry is optional.

Default value

`am-eai-pac-svc`

Example

```
eai-pac-svc-header = am-eai-pac-svc
```

eai-redir-url-header

Use the **eai-redir-url-header** stanza entry to specify the name of the header that contains the URL to which WebSEAL redirects the client after successful authentication.

Syntax

```
eai-redir-url-header = header-name
```

Description

Specifies the name of the header that contains the URL a client is redirected to upon successful authentication.

Options

header-name
The name of the header that contains the URL a client is redirected to upon successful authentication.

Usage

This stanza entry is optional.

Default value

`am-eai-redir-url`

Example

```
eai-redir-url-header = am-eai-redir-url
```

eai-session-id-header

Use the **eai-session-id-header** stanza entry to specify the name of the header that contains the session identifier of the distributed session.

Syntax

```
eai-session-id-header = header-name
```

Description

The name of the header that contains the session identifier of the distributed session to be shared across multiple DNS domains.

Options

header-name
The session identifier of the distributed session to be shared across multiple DNS domains.

Usage

This stanza entry is required.

Default value

`am-eai-session-id`

Example

```
eai-session-id-header = am-eai-session-id
```

eai-user-id-header

Use the **eai-user-id-header** stanza entry to specify the header that contains the user ID for which WebSEAL generates the credential. The specified header must precede all others in the HTTP response.

Syntax

```
eai-user-id-header = header-name
```

Description

Specifies the name of the header that contains the ID of the user used when generating a credential.

Options

header-name
The name of the header that contains the ID of the user used when generating a credential.

Usage

This stanza entry is optional.

Default value

`am-eai-user-id`

Example

```
eai-user-id-header = am-eai-user-id
```

eai-verify-user-identity

Use the **eai-verify-user-identity** stanza entry to control whether the user identity is verified during reauthentication. EAI applications can reauthenticate a user by returning new authentication

information for a previously authenticated session. By default, WebSEAL does not ensure that the new user identity matches the user identity from the previous authentication.

Syntax

```
eai-verify-user-identity = {yes|no}
```

Description

During the EAI re-authentication process, this configuration entry determines whether the new user identity must match the user identity from the previous authentication.

Options

yes
During EAI authentication, the new user identity is compared with the user identity from the previous authentication. If the user identities do not match, an error is returned.

no
EAI authentication proceeds without verifying the new user identity.

Usage

This stanza entry is optional.

Default value

no

Example

```
eai-verify-user-identity = yes
```

eai-xattrs-header

Use the **eai-xattrs-header** stanza entry to specify the HTTP headers in the response from the external authentication interface (EAI) server that contain authentication data. WebSEAL uses these headers to add extended attributes to the credential.

Syntax

```
eai-xattrs-header = header-name[,header-name...]
```

Description

Specifies a comma-delimited list of header names. WebSEAL examines the response for headers with the specified names and creates extended attributes using the name of the header as the attribute name and the value of the header as the attribute value.

For example, if the following headers are returned in the HTTP response:

```
 am-eai-xattrs: creditcardexpiry, streetaddress
 creditcardexpiry: 090812
 streetaddress: 555 homewood lane
```

WebSEAL will:

1. Examine the am-eai-xattrs header
2. Detect two headers to look for in the response

3. Find those headers and their values
4. Add the two specified attributes to the credential

Options

***header-name*[,*header-name*...]**
One or more (comma delimited) header names that are added to the credential as extended attributes.

Usage

This stanza entry is optional.

Default value

`am-eai-xattrs`

Example

```
eai-xattrs-header = am-eai-xattrs
```

retain-eai-session

Syntax

```
retain-eai-session = {yes|no}
```

Description

Specifies whether the existing session and session cache entry for a client are retained or replaced when an already-authenticated EAI client authenticates through an EAI a second time.

Options

yes
If an already-authenticated EAI client authenticates through an EAI a second time, the existing session and session cache entry for the client are retained, and the new credential is stored in the existing cache entry.

no
If an already-authenticated EAI client authenticates through an EAI a second time, the existing session and session cache entry for the client are completely replaced and the new credential is stored in the new cache entry.

Usage

This stanza entry is required.

Default value

no

Example

```
retain-eai-session = no
```

[eai-trigger-urls] stanza

Use the **[eai-trigger-urls]** stanza to specify trigger URL strings for the external authentication interface (EAI). When WebSEAL detects the trigger URL in a request, it intercepts the corresponding response and examines it for authentication data in special HTTP headers.

trigger

Use this **trigger** stanza entry to specify trigger URL strings for standard WebSEAL junctions. WebSEAL examines each server response to these trigger URLs to determine whether the response contains authentication data.

Syntax

```
trigger = url-pattern
```

Description

Format for standard WebSEAL junctions. Specifies the trigger URL that causes WebSEAL to set a special flag on the request. Responses to this request also contain the flag, which causes WebSEAL to intercept and examine the response for authentication data in special HTTP headers.

Options

url-pattern

The trigger URL (format for standard WebSEAL junctions) that causes WebSEAL to set a special flag on the request.

This *url-pattern* is a case-sensitive pattern that can include wildcard characters.

Usage

There must be at least one entry when **eai-auth** is not none.

Default value

None.

Example

```
trigger = /jct/cgi-bin/eaitest/*
```

trigger

Use this **trigger** stanza entry to specify trigger URL strings for virtual host junctions. WebSEAL examines each server response to these trigger URLs to determine whether the response contains authentication data.

Syntax

```
trigger = http[s]://virtual-host-name[:port_number]/url-pattern
```

Description

Format for virtual host junctions. Specifies the trigger URL that causes WebSEAL to set a special flag on the request. Responses to this request also contain the flag, which causes WebSEAL to intercept and examine the response for authentication data in special HTTP headers.

For virtual host junctions to match a trigger, they must use the same protocol and the same *virtual-host-name* and port number as the trigger.

Options

http[s]://*virtual-host-name*[:*port_number*]/*url-pattern*
The trigger URL (format for virtual host junctions) that causes WebSEAL to set a special flag on the request.

Note: The *url-pattern* is case-sensitive. The rest of the trigger is not case-sensitive.

Usage

There must be at least one entry when **eai-auth** is not none.

Default value

None.

Example

```
trigger = https://vhost1.example.com:4344/jct/cgi-bin/eaitest/*
```

[enable-redirects] stanza

Use the **[enable-redirects]** stanza to enable automatic redirection for each of the applicable authentication methods. The authentication methods include forms authentication, basic authentication, certificate authentication, and EAI authentication.

redirect

Syntax

```
redirect = {forms-auth|basic-auth|cert-auth|ext-auth-interface|oidc}
```

Description

Enables redirection for use with one or more authentication mechanism.

Options

{forms-auth|basic-auth|cert-auth|ext-auth-interface|oidc}
Redirection is supported for:

- Forms authentication
- Basic authentication
- Certificate authentication
- External authentication interface
- OpenID Connect (OIDC)

The configuration file must contain a separate entry for each authentication mechanism for which redirection is enabled.

Usage

This stanza entry is optional.

Default value

None.

Example

Example entries that enables redirection for forms authentication and basic authentication:

```
redirect = forms-auth
redirect = basic-auth
```

[failover] stanza

Use the **[failover]** stanza to configure the use of failover cookies in WebSEAL.

clean-ecsso-urls-for-failover

Use the **clean-ecsso-urls-for-failover** stanza entry to control whether the URL that WebSEAL sends during failover authentication includes query arguments that contain the **PD-VFHOST** and **PD-VF** tokens.

Syntax

```
clean-ecsso-urls-for-failover = {yes|no}
```

Description

You can enable Failover Authentication and eCSSO in your environment. During failover authentication, if a user was originally authenticated using eCSSO, WebSEAL updates the URL that it sends to the back-end server. WebSEAL sends **PD-VFHOST** and **PD-VF** tokens as query arguments, along with the original URL.

Use the **clean-ecsso-urls-for-failover** configuration entry to control whether these tokens are removed from the URL.

Options

yes
The query arguments that contain the PD-VFHOST and PD-VF tokens are removed from the URL.

no
The query arguments that contain the PD-VFHOST and PD-VF tokens are not removed from the URL.

Usage

This stanza entry is optional.

Default value

no

Example

```
clean-ecsso-urls-for-failover = no
```

enable-failover-cookie-for-domain

Use the **enable-failover-cookie-for-domain** stanza entry to enable the failover cookie for the domain. When enabled, the failover authentication cookie can be used by any WebSEAL server that is in the same domain as the WebSEAL server that creates the cookie.

Syntax

```
enable-failover-cookie-for-domain = {yes|no}
```

Description

Enables the failover cookie for the domain.

Options

yes
Enables the failover cookie for the domain.

no
Disables the failover cookie for the domain.

Usage

This stanza entry is required.

Default value

no

Example

```
enable-failover-cookie-for-domain = no
```

failover-auth

Use the **failover-auth** stanza entry to enable failover cookies. The configured value specifies the protocol over which WebSEAL accepts cookies for authentication during a failover authentication event.

Syntax

```
failover-auth = {none|http|https|both}
```

Description

Enables WebSEAL to accept failover cookies.

Options

{none|http|https|both}
Specifies which protocols are supported. The value `both` means both HTTP and HTTPS.

Usage

This stanza entry is required.

Default value

none

Example

```
failover-auth = none
```

failover-cookie-lifetime

Use the **failover-cookie-lifetime** stanza entry to specify the lifetime, in minutes, of the failover cookie contents.

Syntax

```
failover-cookie-lifetime = number_of_minutes
```

Description

An integer value specifying the number of minutes that failover cookie contents are valid.

Options

number_of_minutes
An integer value specifying the number of minutes that failover cookie contents are valid. Must be a positive integer. There is no maximum value.

Usage

This stanza entry is required.

Default value

60

Example

```
failover-cookie-lifetime = 60
```

failover-cookie-name

Use the **failover-cookie-name** stanza entry to specify the name of the cookie that stores the failover token. If the WebSEAL server that is hosting the user session fails, this cookie is presented to a replicated WebSEAL server to automatically reauthenticate the user.

Syntax

```
failover-cookie-name = <cookie_name>
```

Description

The name of the cookie that WebSEAL uses to store the failover token. The token includes details of the user session, which WebSEAL can use for single sign-on to a different WebSEAL instance in the event of failover.

Options

cookie_name
The name of the failover cookie.

Usage

This stanza entry is optional.

Default value

PD-ID

Example

```
failover-cookie-name = fail_ckie
```

failover-cookies-keyfile

Use the **failover-cookies-keyfile** stanza entry to specify the key file that WebSEAL can use to encrypt and decrypt the data in failover cookies.

Syntax

```
failover-cookies-keyfile = file_name
```

Description

A key file for failover cookie encryption. Use the SSO Keys management page of the LMI to generate this file.

Options

file_name
Name of the key file for failover cookie encryption.

Usage

This stanza entry is optional.

Default value

None.

Example

```
failover-cookies-keyfile = failover.key
```

failover-include-session-id

Use the **failover-include-session-id** stanza entry to enable WebSEAL to reuse the original session ID of a client by storing it as an attribute in the failover cookie. Reusing session IDs can improve failover authentication response and performance in a non-sticky load-balancing environment.

Syntax

```
failover-include-session-id = {yes|no}
```

Description

Enable or disable WebSEAL to reuse a client's original session ID to improve failover authentication response and performance in a non-sticky load-balancing environment. WebSEAL reuses the original session ID by storing the ID as an extended attribute to the failover cookie.

Options

yes
Enable WebSEAL to reuse a client's original session ID to improve failover authentication response and performance in a non-sticky load-balancing environment.

no
Disable WebSEAL to reuse a client's original session ID to improve failover authentication response and performance in a non-sticky load-balancing environment.

Usage

This stanza entry is required.

Default value

no

Example

```
failover-include-session-id = no
```

failover-require-activity-timestamp-validation

Use the **failover-require-activity-timestamp-validation** stanza entry to control whether WebSEAL requires each failover authentication cookie to contain a session activity timestamp. This stanza entry is used primarily for compatibility with prior versions of WebSEAL.

Syntax

```
failover-require-activity-timestamp-validation = {yes|no}
```

Description

Enables or disables the requirement of a session activity timestamp validation in the failover cookie.

Options

yes
Enables the requirement of a session activity timestamp validation in the failover cookie.

no
Disables the requirement of a session activity timestamp validation in the failover cookie. For backward compatibility with versions of WebSEAL server prior to version 5.1, set this stanza entry to no. Versions prior to version 5.1 did not create the session activity timestamp in the failover cookie.

Usage

This stanza entry is required.

Default value

no

Example

```
failover-require-activity-timestamp-validation = no
```

failover-require-lifetime-timestamp-validation

Use the **failover-require-lifetime-timestamp-validation** stanza entry to control whether WebSEAL requires each failover authentication cookie to contain a session lifetime timestamp. This stanza entry is used primarily for compatibility with prior versions of WebSEAL.

Syntax

```
failover-require-lifetime-timestamp-validation = {yes|no}
```

Description

Enables or disables the requirement of a session lifetime timestamp validation in the failover cookie.

Options

yes
Enables the requirement of a session lifetime timestamp validation in the failover cookie.

no
Disables the requirement of a session lifetime timestamp validation in the failover cookie. For backward compatibility with versions of WebSEAL server prior to version 5.1, set this stanza entry to no. Versions prior to version 5.1 did not create the session lifetime timestamp in the failover cookie.

Usage

This stanza entry is required.

Default value

no

Example

```
failover-require-lifetime-timestamp-validation = no
```

failover-update-cookie

Use the **failover-update-cookie** stanza entry to specify the frequency at which WebSEAL updates the session activity timestamp in failover cookies.

Syntax

```
failover-update-cookie = number_of_seconds
```

Description

The maximum interval, in number of seconds, allowed between updates of the session activity timestamp in the failover cookies. The value is an integer. When the server receives a request, if the number of seconds specified for this parameter has passed, the session activity timestamp is updated.

Note: If you set a value greater than zero, ensure that you also set `session-activity-timestamp = add` in the **[failover-add-attributes]** stanza. If you do not set this additional value, WebSEAL decodes the failover cookie on each user access, which can adversely affect performance.

Options

number_of_seconds
When the value is 0, the session activity timestamp is updated on every request. When the value is less than zero (negative number), the session activity timestamp is never updated. There is no maximum value.

Usage

This stanza entry is required.

Default value

-1

Example

```
failover-cookie-update = 60
```

reissue-missing-failover-cookie

Syntax

```
reissue-missing-failover-cookie = {yes|no}
```

Description

Allows WebSEAL to reissue a cached original failover cookie in the response to a client, if the client makes a request that does not include the failover cookie.

Options

yes
Enables the failover cookie reissue mechanism.

no
Disables the failover cookie reissue mechanism.

Usage

This stanza entry is required.

Default value

no

Example

```
reissue-missing-failover-cookie = no
```

use-utf8

Syntax

```
use-utf8 = {yes|no}
```

Description

Use UTF–8 encoding for strings in the failover authentication cookie.

Options

yes
Beginning with version 5.1, WebSEAL servers use UTF-8 encoding by default. When this stanza entry is set to yes, cookies can be exchanged with other WebSEAL servers that use UTF-8 encoding. This enables cookies to used across different code pages (such as for a different language).

no
For backward compatibility with cookies created by WebSEAL servers from version prior to 5.1, set this stanza entry to no.

Usage

This stanza entry is required.

Default value

yes

Example

```
use-utf8 = yes
```

[failover-add-attributes] stanza

Use the **[failover-add-attributes]** stanza to add attributes to the failover authentication cookie. These attributes can include extended attributes from a user credential and session time stamps.

attribute_pattern

Use the **attribute_pattern** stanza entry to specify the credential attributes that WebSEAL preserves in the failover cookie.

Syntax

```
attribute_pattern = add
```

Description

List of attributes from the original credential that must be preserved in the failover cookie.

The order of entries in the stanza is important. Rules (patterns) that are listed earlier in the stanza take precedence over those entries that are listed later in the stanza. Attributes that do not match any pattern are not added to the failover cookie.

Options

attribute_pattern
The attribute pattern is a not case-sensitive wildcard pattern.

add
Add attribute.

Usage

Entries in this stanza are optional.

Default value

There are no default entries in this stanza. However, the attributes AUTHENTICATION_LEVEL and AZN_CRED_AUTH_METHOD are added to the failover cookie by default. You do not need to include these attributes in the configuration stanza.

Example

```
tagvalue_failover_amweb_session_id = add
```

session-activity-timestamp

Syntax

```
session-activity-timestamp = add
```

Description

This entry specifies that the timestamp for the last user activity be taken from the failover cookie and added to the new session on the replicated server.

This attribute cannot be specified by pattern matching. This entry must be added exactly as it is written.

Options

add
Add attribute.

Usage

This stanza entry is optional and must be manually added to the configuration file.

Default value

None.

Example

```
session-activity-timestamp = add
```

session-lifetime-timestamp

Syntax

```
session-lifetime-timestamp = add
```

Description

This entry specifies that the timestamp for creation of the original session be taken from the failover cookie and added to the new session on the replicated server.

This attribute cannot be specified by pattern matching. This entry must be added exactly as it is written.

Options

add
Add attribute.

Usage

This stanza entry is optional and must be manually added to the configuration file.

Default value

None.

Example

```
session-lifetime-timestamp = add
```

[failover-restore-attributes] stanza

Use the **[failover-restore-attributes]** stanza to configure WebSEAL to extract certain attributes from the failover authentication cookie and place them into the user credential.

attribute_pattern

Use the *attribute_pattern* stanza entry with a value of preserve to specify the failover cookie attributes that WebSEAL adds to the re-created user credential.

Syntax

```
attribute_pattern = preserve
```

Description

List of attributes to put in the new credential when re-creating a credential from a failover cookie.

The order of entries in the stanza is important. Rules (patterns) that are listed earlier in the stanza take precedence the entries that are listed later in the stanza. Attributes that do not match any pattern are not added to the credential.

Options

attribute_pattern
The attribute pattern is a not case-sensitive wildcard pattern.

preserve
When WebSEAL recreates a credential, all failover cookie attributes are ignored unless specified by an entry with the value `preserve`

Usage

Entries in this stanza are optional.

Default value

None.

Example

```
tagvalue_failover_amweb_session_id = preserve
```

attribute_pattern

Use the ***attribute_pattern*** stanza entry with a value of `refresh` to specify failover cookie attributes that WebSEAL must refresh rather than preserve.

Syntax

```
attribute_pattern = refresh
```

Description

A list of failover cookie attributes to omit from the recreated user credential.

This list is not needed in all configurations. The default behavior when WebSEAL recreates a user credential is to omit all attributes that are not specified with a value of `preserve`.

In some cases, it might be necessary to specify an exception to a wildcard pattern matching to ensure that a specific attribute gets refreshed, not preserved. To do so, configure the pattern with the value `refresh`. This specification might be necessary, for example, when you are using a custom external authentication C API module.

The order of entries in the stanza is important. Rules (patterns) that are listed earlier in the stanza take precedence over the entries that are listed later in the stanza.

Options

attribute_pattern
: The attribute pattern is a not case-sensitive wildcard pattern.

refresh
: Specifies an exception to a wildcard pattern matching to ensure that a specific attribute gets refreshed, not preserved.

Usage

Entries in this stanza are optional.

Default value

None.

Example

```
tagvalue_failover_amweb_session_id = refresh
```

[filter-advanced-encodings] stanza

Use the **[filter-advanced-encodings]** stanza to configure the types of URL encoding that are detected and filtered.

Syntax

```
[filter-advanced-encodings]
     <escaping method> = <chars to escape>
```

```
<escaping method> = <chars to escape>
...
```

Description

The WebSEAL advanced filtering can process a number of URL encoding types. Use this stanza to define the types of encoding to be detected and filtered.

Options

<escaping method>

ampersand
Ampersand encoded. For example:

```
HTTP://host:port/path?V1=D1&V2=D2
```

ampersand-hex
Ampersand just hex encoded. For example:

```
HTTP&amp;#x3a;&amp;#x2f;&amp;#x2f;host&amp;#x3a;port&amp;#x2f;
```

ampersand-dec
Ampersand just dec encoded. For example:

```
HTTP&amp;#58;&amp;#57;&amp;#57;host&amp;#58;port&amp;#57;
```

escaped
Backslash encoded. For example:

```
HTTP:\/\/host:port\/
```

percent
Percent hex encoded. For example:

```
HTTP%3A%2F%2Fhost%3Aport%2F
```

escaped-u
Backslash U hex encoded. For example:

```
HTTP:\u002f\u002fhost:port\u002f
```

percent-u
Percent U hex encoded. For example:

```
HTTP%u003a//host%u003aport/
```

escaped-x
Backslash X hex encoded. For example:

```
HTTP\x3A\x2F\x2Fhost\x3Aport\x2F
```

<chars to escape>
A list of characters that need encoding, which are governed by the following rules:

- If two characters are separated by a '-' (hyphen) character, then this is a range of characters to encode. For example, "A-Z" indicates all characters from 'A' to 'Z' including 'A' and 'Z'.
- If the first character in the list is the '^' character, then the list of characters are those not to encode. For example "^A-Za-z" indicates all characters excluding characters from 'A' to 'Z' and excluding characters from 'a' to 'z'.
- If the first character (excluding the '^' character) is a '-' (hyphen) character, then that is taken as the literal '-' character rather than representing a range of characters.

Usage

It is permissible to have multiple entries with the same *<escaping method>*, if they produce different encodings of the "`://`" string.

WebSEAL uses the *<escaping method>* against *<chars to escape>* to encode the string "`://`" and use that encoded value in combination with "`http`" or "`https`" to detect encoded URLs. The very first entry should be to define the "ampersand" encoding method and not list the character '`:`' and '`/`' in the *<chars to escape>*. This then matches URLs with an un-encoded "`://`".

Default value

None.

Example

```
[filter-advanced-encodings]
ampersand     = &<>"'
ampersand-hex = ^a-zA-Z0-9.
ampersand-dec = ^a-zA-Z0-9.
percent       = ^a-zA-Z0-9.
escaped-x     = ^a-zA-Z0-9.
```

This example specifies the following behavior:

`ampersand = &<>"'`
: This allows WebSEAL to find and filter unencoded links such as "`http://backend.com:80/`". It identifies the link by looking for "`http`" or "`https`" followed by "`://`". Any WebSEAL host name or junction path replaced in the filtered link will have the characters &<>"' replaced by their encoded forms, `&` `<` `>` `&quot;` and `&#x2c;` respectively.

`ampersand-hex = ^a-zA-Z0-9.`
: This allows WebSEAL to find and filter ampersand hex encoded links such as "`http&#x3a;&#x2f;&#x2f;backend.com&#x3a;80&#x2f;`". It identifies the link embedded in java script by looking for "`http`" or "`https`" followed by "`&#x3a;&#x2f;&#x2f;`". Any WebSEAL host name or junction path replaced in the filtered link will have the characters not in the set `a-zA-Z0-9.` replaced by their encoded forms `&#xHH;`.

`ampersand-dec = ^a-zA-Z0-9.`
: This allows WebSEAL to find and filter ampersand hex encoded links such as "`http&#58;&#47;&#47;backend.com&#58;80&#47;`". It identifies the link embedded in java script by looking for "`http`" or "`https`" followed by "`&#58;&#47;&#47;`". Any WebSEAL host name or junction path replaced in the filtered link will have the characters not in the set `a-zA-Z0-9.` replaced by their encoded forms `&#DDD;`.

`percent = ^a-zA-Z0-9.`
: This allows WebSEAL to find and filter ampersand hex encoded links such as "`http%3a%2f%2fbackend.com%3a80%2f`". It identifies the link embedded in java script by looking for "`http`" or "`https`" followed by "`%3a%2f%2f`". Any WebSEAL host name or junction path replaced in the filtered link will have the characters not in the set `a-zA-Z0-9.` replaced by their encoded forms %HH. This may be required for attributes with Flash URLs.

`escaped-x = ^a-zA-Z0-9.`
: This allows WebSEAL to find and filter ampersand hex encoded links such as "`http\x3a\x2f\x2fbackend.com\x3a80\x2f`". It identifies the link embedded in java script by looking for "`http`" or "`https`" followed by "`\x3a\x2f\x2f`". Any WebSEAL host name or junction path replaced in the filtered link will have the characters not in the set `a-zA-Z0-9.` replaced by their encoded forms \xHH. This may be required for Javascript encoded URLs.

[filter-content-types] stanza

Use the **`[filter-content-types]`** stanza to specify the content (MIME) types of the documents in which WebSEAL filters tag-based static URLs.

type

Syntax

```
type = type_name
```

Description

List of entries that specify MIME types to be filtered by WebSEAL when received from junctioned servers.

Administrators can add additional MIME types that refer to a document that contains HTML or HTML-like content.

Options

type_name
MIME type.

Usage

This list of stanza entries is required.

Default value

Do not remove the default entries.

```
type = text/html
type = text/vnd.wap.wml
```

Example

```
type = text/html
type = text/vnd.wap.wml
```

[filter-events] stanza

Use the **[filter-events]** stanza to identify HTML tags that might contain JavaScript. WebSEAL searches these tags to filter any absolute URLs embedded in JavaScript event handlers.

HTML_tag

Syntax

```
HTML_tag = event_handler
```

Description

List of HTML tags used by WebSEAL to identify and filter absolute URLs embedded in JavaScript. JavaScript allows HTML tags to contain *event handlers* that are invoked when certain events occur. For example, the HTML tag:

```
<form onsubmit="javascript:doSomething()">
```

causes the JavaScript function `doSomething()` to be called when the form is submitted.

The entries in this stanza are used to identify HTML tags that may contain JavaScript code. When such a tag is discovered, WebSEAL searches the tag to filter any absolute URLs embedded in the JavaScript. For example, if the "form onsubmit" example looked like:

```
<form onsubmit="javaScript:doSomething('http://junction.server.com')">
```

WebSEAL HTML filtering would modify the tag to look like:

```
<form onsubmit="javaScript:doSomething('/junction')">
```

Administrators can add additional entries when necessary. New entries must consist of valid HTML tags that are built into JavaScript. When adding new entries, maintain alphabetical order.

Options

HTML_tag
HTML tag.

event_handler
JavaScript event handler.

Usage

This list is required. Although not all tags are required by all applications, the unused tags do no harm. Leave the default entries in this list.

Default value

Default HTML tags and event handlers:

```
A = ONCLICK
A = ONDBLCLICK
A = ONMOUSEDOWN
A = ONMOUSEOUT
A = ONMOUSEOVER
A = ONMOUSEUP
AREA = ONCLICK
AREA = ONMOUSEOUT
AREA = ONMOUSEOVER
BODY = ONBLUR
BODY = ONCLICK
BODY = ONDRAGDROP
BODY = ONFOCUS
BODY = ONKEYDOWN
BODY = ONKEYPRESS
BODY = ONKEYUP
BODY = ONLOAD
BODY = ONMOUSEDOWN
BODY = ONMOUSEUP
BODY = ONMOVE
BODY = ONRESIZE
BODY = ONUNLOAD
FORM = ONRESET
FORM = ONSUBMIT
FRAME = ONBLUR
FRAME = ONDRAGDROP
FRAME = ONFOCUS
FRAME = ONLOAD
FRAME = ONMOVE
```

```
FRAME = ONRESIZE
FRAME = ONUNLOAD
IMG = ONABORT
IMG = ONERROR
IMG = ONLOAD
INPUT = ONBLUR
INPUT = ONCHANGE
INPUT = ONCLICK
INPUT = ONFOCUS
INPUT = ONKEYDOWN
```

```
INPUT = ONKEYPRESS
INPUT = ONKEYUP
INPUT = ONMOUSEDOWN
INPUT = ONMOUSEUP
INPUT = ONSELECT
LAYER = ONBLUR
LAYER = ONLOAD
LAYER = ONMOUSEOUT
LAYER = ONMOUSEOVER
SELECT = ONBLUR
SELECT = ONCHANGE
SELECT = ONFOCUS
TEXTAREA = ONBLUR
TEXTAREA = ONCHANGE
TEXTAREA = ONFOCUS
TEXTAREA = ONKEYDOWN
TEXTAREA = ONKEYPRESS
TEXTAREA = ONKEYUP
TEXTAREA = ONSELECT
```

Example

```
IMG = ONABORT
```

[filter-request-headers] stanza

Use the **[filter-request-headers]** stanza to configure extra HTTP headers for WebSEAL to filter before it sends a request to a junctioned server.

header

Syntax

```
header = header_name
```

Description

List of HTTP headers that WebSEAL filters before sending the request to a junctioned server. A default list is built-in to WebSEAL. The default entries are not included in the configuration file.

The addition of new entries in this stanza is optional. For example, an administrator could add the **accept-encoding** header. This would instruct WebSEAL to remove any **accept-encoding** headers from requests before forwarding the request to the junction. The removal of the **accept-encoding** header would cause the junction server to return the document in an unencoded form, allowing WebSEAL to filter the document if necessary.

New entries must consist of valid HTTP headers.

This configuration item may be customized for a particular junction by adding the adjusted configuration item to a `[junction:{jct_id}]` stanza, where `'{jct-id}'` refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

Options

header_name
HTTP header name.

Usage

The addition of new entries in this stanza is optional.

Default value

Default built-in header list:

```
host
connection
proxy-connection
expect
te
iv-ssl-jct
iv-user
iv_user
iv-groups
iv_groups
iv-creds
iv_creds
iv_remote_address
iv-remote-address
```

Example

```
header = accept-encoding
```

[filter-request-headers:<jct-id>] stanza

header

Syntax

```
header = header_name
```

Description

List of HTTP headers that WebSEAL filters before sending the request to a junctioned server. A default list is built-in to WebSEAL. The default entries are not included in the configuration file.

The addition of new entries in this stanza is optional. For example, an administrator could add the **accept-encoding** header. This would instruct WebSEAL to remove any **accept-encoding** headers from requests before forwarding the request to the junction. The removal of the **accept-encoding** header would cause the junction server to return the document in an unencoded form, allowing WebSEAL to filter the document if necessary.

New entries must consist of valid HTTP headers.

This configuration item may be customized for a particular junction by adding the adjusted configuration item to a `[junction:{jct_id}]` stanza, where `'{jct-id}'` refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

Options

header_name
HTTP header name.

Usage

The addition of new entries in this stanza is optional.

Default value

Default built-in header list:

```
host
connection
proxy-connection
expect
te
iv-ssl-jct
iv-user
iv_user
iv-groups
iv_groups
iv-creds
iv_creds
iv_remote_address
iv-remote-address
```

Example

```
header = accept-encoding
```

[filter-schemes] stanza

Use the **[filter-schemes]** stanza to list URL schemes that are not to be filtered by WebSEAL in responses from junctioned application servers.

scheme

Syntax

```
scheme = scheme_name
```

Description

List of URL schemes that are *not* to be filtered by WebSEAL. A scheme is a protocol identifier.

This list is used when WebSEAL encounters a document containing a base URL. For example:

```
<head>
<base href="http://www.foo.com">
</head>
<a href="mailto:bee@bee.com>Send me mail",/a>
```

WebSEAL identifies the scheme `mailto` because this scheme is included by default in the **[filter-schemes]** stanza. If `mailto` was not identified as a scheme, WebSEAL would interpret it as document and perform normal filtering. WebSEAL would then rewrite the link as:

```
<a href="http://www.foo.com/mailto:bee@bee.com"
```

This would be incorrect.

Options

scheme_name
Scheme name.

Usage

WebSEAL provides a set of default schemes. The administrator can extend the list if additional protocols are used. Do not delete entries from the list.

Default value

Default list entries:

```
scheme = file
scheme = ftp
scheme = https
scheme = mailto
scheme = news
scheme = telnet
```

Example

```
scheme = telnet
```

[filter-url] stanza

Use the **[filter-url]** stanza to specify the HTML tags and attributes that WebSEAL filters in responses from junctioned servers.

HTML_tag

Syntax

```
HTML_tag = URL_attribute
```

Description

List of URL attributes that WebSEAL server filters in responses from junctioned servers.

Administrators can add additional entries when necessary. New entries must consist of valid HTML tags and attributes. When adding new entries, maintain alphabetical order.

Options

URL_attribute
URL attribute.

Usage

This list is required. Although not all tags are required by all applications, the unused tags do no harm. Leave the default entries in this list.

Default value

Default HTML tags and attributes:

```
A = HREF
APPLET = CODEBASE
AREA = HREF
BASE = HREF
BGSOUND = SRC
BLOCKQUOTE = CITE
BODY = BACKGROUND
DEL = CITE
```

```
DIV = EMPTYURL
DIV = IMAGEPATH
DIV = URL
DIV = VIEWCLASS
EMBED = PLUGINSPAGE
EMBED = SRC
FORM = ACTION
FRAME = LONGDESC
FRAME = SRC
HEAD = PROFILE
IFRAME = LONGDESC
IFRAME = SRC
ILAYER = BACKGROUND
ILAYER = SRC
IMG = SRC
IMG = LOWSRC
IMG = LONGDESC
IMG = USEMAP
IMG = DYNSRC
```

```
INPUT = SRC
INPUT = USEMAP
INS = CITE
ISINDEX = ACTION
ISINDEX = HREF
LAYER = BACKGROUND
LAYER = SRC
LINK = HREF
LINK = SRC
OBJECT = CODEBASE
OBJECT = DATA
OBJECT = USEMAP
Q = CITE
SCRIPT = SRC
TABLE = BACKGROUND
TD = BACKGROUND
TH = BACKGROUND
TR = BACKGROUND
WM:CALENDARPICKER = FOLDERURL
WM:CALENDARPICKER = IMAGEPREVARROW
WM:CALENDARPICKER = IMAGENEXTARROW
WM:CALENDARVIEW = FOLDERURL
WM:MESSAGE = DRAFTSURL
WM:MESSAGE = URL
WM:NOTIFY = FOLDER
WM:REMINDER = FOLDER
?IMPORT = IMPLEMENTATION
```

Example

```
IMG = SRC
```

[flow-data] stanza

Use the **[flow-data]** stanza to configure the recording of flow data statistics in WebSEAL.

flow-data-enabled

Use the **flow-data-enabled** stanza entry to control whether WebSEAL records flow data statistics.

Syntax

```
flow-data-enabled = {yes|no}
```

Description

The appliance can record statistical information about incoming WebSEAL requests. Use this parameter to enable or disable the recording of flow data statistics.

If you set this parameter to yes, you can also use the **flow-data-stats-interval** parameter in the **[flow-data]** stanza to set the frequency for gathering statistics.

Note: You can configure the **[user-agent]** stanza to categorize the incoming user-agent requests and make the statistical data more useful. You can then view a statistical breakdown of all requests that is based on user-agent and junction.

Options

yes
WebSEAL records statistics about incoming requests.

no
WebSEAL does not record statistics about incoming requests.

Usage

This stanza entry is optional.

Default value

yes

Example

```
flow-data-enabled = yes
```

flow-data-stats-interval

Use the **flow-data-stats-interval** stanza entry to control how frequently the appliance collects flow data statistics.

Syntax

```
flow-data-stats-interval = number_of_seconds
```

Description

This parameter specifies the statistics interval in seconds. At each time interval, WebSEAL records statistical information about incoming requests. The default value of 600 records statistics every 10 minutes.

To gather statistics at the specified interval, you must use the **flow-data-enabled** parameter, also in the **[flow-data]** stanza, to enable the flow data statistics on the appliance.

Note: You can configure the **[user-agent]** stanza to categorize the incoming user-agent requests and make the statistical data more meaningful. You can then view a statistical breakdown of all requests that is based on user-agent and junction.

Options

number_of_seconds
Specifies the interval that the appliance uses to collect flow data statistics.

Usage

This stanza entry is optional.

Default value

600

Example

```
flow-data-stats-interval = 600
```

[forms] stanza

Use the **[forms]** stanza to configure forms authentication in WebSEAL.

allow-empty-form-fields

Use the **allow-empty-form-fields** stanza entry to determine whether WebSEAL returns an error for login requests that contain an empty user name or an empty password.

Syntax

```
allow-empty-form-fields = {true|false}
```

Description

If a forms login request is received with either an empty user name or an empty password, WebSEAL returns the login form without stating an error.

If you prefer that an error message is displayed with the returned login form, set this value to `true`. In this case, WebSEAL attempts to authenticate the user and if the values have zero length, the registry returns the appropriate error.

Options

true
Error message is displayed with the returned login form.

false
Error message is not displayed with the returned login form.

Usage

This stanza entry is required.

Default value

`false`

Example

```
allow-empty-form-fields = false
```

forms-auth

Use the **forms-auth** stanza entry to enable the forms authentication method.

Syntax

```
forms-auth = {none|http|https|both}
```

Description

Enables authentication using the Forms Authentication mechanism.

Options

{none|http|https|both}
Specifies which protocols are supported. The value `both` means both HTTP and HTTPS.

Usage

This stanza entry is required.

Default value

none

Example

```
forms-auth = none
```

[gso-cache] stanza

Use the **[gso-cache]** stanza to define Global Signon (GSO) settings.

gso-cache-enabled

Use the **gso-cache-enabled** stanza entry to enable or disable Global Signon (GSO) cache.

Syntax

```
gso-cache-enabled = {yes|no}
```

Description

Enables or disables the Global Signon (GSO) cache.

Options

yes
Enables the Global Signon (GSO) cache.

no
Disables the Global Signon (GSO) cache.

Usage

This stanza entry is required.

Default value

no

Example

```
gso-cache-enabled = no
```

gso-cache-entry-idle-timeout

Use the **gso-cache-entry-idle-timeout** stanza entry to specify the timeout for cache entries that are idle.

Syntax

```
gso-cache-entry-idle-timeout = number_of_seconds
```

Description

Integer value that specifies the timeout, in seconds, for cache entries that are idle.

Options

number_of_seconds
The value must be greater than or equal to zero (0). A value of 0 means that entries are not removed from the GSO cache due to inactivity. However, they may still be removed due to either the **gso-cache-size** being exceeded or the **gso-cache-entry-lifetime** stanza entry being exceeded. WebSEAL does not impose a maximum value.

Usage

This stanza entry is required, but is ignored when GSO caching is disabled.

Default value

120

Example

```
gso-cache-entry-idle-timeout = 120
```

gso-cache-entry-lifetime

Use the **gso-cache-entry-lifetime** stanza entry to define the lifetime of a GSO cache entry.

Syntax

```
gso-cache-entry-lifetime = number_of_seconds
```

Description

Integer value that specifies the lifetime, in seconds, of a GSO cache entry.

Options

number_of_seconds
The value must be greater than or equal to zero (0). A value of 0 means that entries are not removed from the GSO cache due to their entry lifetime being exceeded. However, they may still be removed due to either the **gso-cache-size** being exceeded or the **gso-cache-entry-idle-timeout** stanza entry being exceeded. WebSEAL does not impose a maximum value.

Usage

This stanza entry is required, but is ignored when GSO caching is disabled.

Default value

900

Example

```
gso-cache-entry-lifetime = 900
```

gso-cache-size

Use the **gso-cache-size** stanza entry to specify how many entries are allowed in the GSO cache.

Syntax

```
gso-cache-size = number_of_entries
```

Description

Integer value indicating the number of entries allowed in the GSO cache.

Options

number_of_entries

The value must be greater than or equal to zero (0). Zero means that there is no limit on the size of the GSO cache. This is not recommended.

WebSEAL does not impose a maximum value. Choose your maximum value to stay safely within the bounds of your available system memory.

Usage

This stanza entry is required, but is ignored when GSO caching is disabled.

Default value

1024

Example

```
gso-cache-size = 1024
```

[header-names] stanza

Use the **[header-names]** stanza to define the HTTP header information in the request that WebSEAL sends to junctioned applications.

header-data

Use the *header-data* stanza entry to add HTTP headers to the request that WebSEAL sends to junctioned applications.

Syntax

```
<header-data> = [+]<header-name>
```

Description

Controls the addition of HTTP headers into the request that is passed to junctioned applications.

To include the same *<header-data>* in different headers, specify multiple entries with the same *<header-data>* value.

Options

<header-data>

The type of data that WebSEAL adds to the *<header-name>* header of the request. The valid values for this entry are as follows:

server-name
The Security Verify Access authorization server name for the WebSEAL server. This name is the name of the authorization API administration server that is used in the **server task** commands.

client-ip-v4
The IPv4 address of the client of this request.

client-ip-v6
The IPv6 address of the client of this request.

client-port
The port that is used by the client of this request. This port is the client source port and not the destination port.

host-name
The host name of the WebSEAL server. WebSEAL obtains this host name from the **web-host-name** configuration entry in the **[server]** stanza if specified. Otherwise, WebSEAL returns the host name of the server itself.

httphdr{*<name>*}
An HTTP header from the request as specified by the *<name>* field. If the HTTP header is not found in the request, WebSEAL uses the value in the **[server] tag-value-missing-attr-tag** configuration entry as the value for the header.

text{<value>}
The literal value which is to be assigned to the specified header.

credattr{<name>}
An attribute from the user's credential, as specified by the <name> field. If the specified attribute does not exist, the value contained within the **[server] tag-value-missing-attr-tag** configuration entry will be used as the value for the header. If the specified attribute is a multi-valued attribute the values will be added to a single HTTP header, with each value separated by a comma.

<header-name>

The name of the HTTP header that holds the data. The *<header-name>* can be prefixed with the plus (+) character if you want to append to any existing header instead of overwriting the existing header. Valid strings are limited to the following characters: A-Z, a-z, 0–9, hyphen (-), or underscore (_).

Usage

This stanza entry is optional.

Default value

`server-name = iv_server_name`

Example

```
server-name = iv_server_name
```

In this example, WebSEAL passes the following header and value to the junction if the WebSEAL instance is `default-webseald-diamond.example.com`:

```
iv_server_name:default-webseald-diamond.example.com
```

Other example entries:

```
client-ip-v4 = +X-Forwarded-For
client-ip-v4 = X-Header
httphdr{host} = X-Forwarded-Host
host-name = X-Forwarded-Server
text{green} = X-Deployment-Status
credattr{AZN_CRED_PRINCIPAL_NAME} = X-Principal
```

[http-method-perms] stanza

Use this stanza to define the permissions that are required to perform a request with a particular HTTP method.

This stanza can be specified for specific junctions by creating a stanza of the form `[http-method-perms:`*`junction`*`]`. The values in the global **[http-method-perms]** stanza apply to any junctions that do not have a junction-specific stanza. A junction-specific stanza **[http-method-perms:junction]** does not inherit values from a global stanza. If the **[http-method-perms:junction]** stanza is defined for a junction, the global **[http-method-perms]** stanza will have no impact.

http-method

Use the *http-method* stanza entry to define the ACL permission for a particular HTTP method.

Syntax

```
http-method = permission
```

Description

The *http-method* entry defines the mapping of an HTTP request method to an ACL permission bit.

Options

- T
- c
- g
- m
- d
- b
- s
- v
- a
- B
- t
- R
- r
- x
- l

- N
- W
- A

You can also create custom permissions in custom action groups, for example, **[my-action-group]t**. See 'Custom permissions in custom action groups' in the Manage access control section of the "Access Manager Platform and Supporting Components Administration Topics".

Usage

This stanza entry is optional. If this stanza is empty, the Web Reverse Proxy defaults to the following ACL bits:

```
PUT => m
DELETE => d
All else (GET,POST .. ) => r
```

The <default> entry defines the permissions that are required for any methods that are not explicitly specified elsewhere in this stanza. A **<default>** entry is required in each **[http-method-perms]** stanza if any methods are defined in the stanza. In other words, an **[http-method-perms]** stanza can either be empty or contains entries that include the **<default>** method.

You can define multiple ACL bits for a method. For example, **POST = Ax**.

If multiple ACL bits are defined for a method, then the user must be granted each of those permissions in the effective ACL for the protected object.

Default value

None. By default, this stanza is not defined.

Example

```
<default> = r
GET = r
HEAD = T
PUT = m
POST = Ax
DELETE = d
TRACE = [my-action-group]t
```

[http-transformations] stanza

Use the **[http-transformations]** stanza to define the HTTP transformation settings.

resource-name

Use the **resource-name** stanza entry to define HTTP transformation resources.

Syntax

```
resource-name = resource-file
```

Description

This configuration information is necessary to support WebSEAL HTTP transformations. You can use WebSEAL HTTP transformations to modify HTTP requests and HTTP responses (excluding the HTTP body) using XSLT or Lua scripting.

WebSEAL will use the extension of the resource file name to determine whether the resource file corresponds to a Lua script, or an XSLT file. Any resource file names which end with '.lua' will be treated as a Lua script, and all other files will be treated as an XSLT file.

For more details, see HTTP transformations.

Options

resource-name
The name of the HTTP transformation resource.

resource-file
The name of the resource file.

Note: You must restart WebSEAL for changes to a rules file to take effect.

Usage

This stanza entry is optional.

Comments

If an HTTP transformation rule modifies the URI or host header of the request, WebSEAL reprocesses the transformed request. This reprocessing ensures that the transformation does not bypass WebSEAL authorization. This behavior also means that administrators can define HTTP transformations rules to send requests to different junctions.

Note: WebSEAL performs reprocessing (and authorization) on the first HTTP transformation only. WebSEAL does not reprocess the new requests that result from these subsequent transformations.

Default value

None.

Example

```
resourceOne = resourceOne.xsl
resourceTwo = resourceTwo.lua
```

[http-transformations:<resource-name>] stanza

Use this stanza to house configuration that is specific to a particular HTTP transformation resource. The <resource-name> component of the stanza name must be changed to the actual name of the resource.

cred-attr-name

Use the **cred-attr-name** stanza entry to define the credential attribute that is included in the XML input document and used when evaluating the HTTP transformation rule.

Syntax

```
cred-attr-name = name
```

Description

This configuration entry can be specified multiple times if multiple credential attributes are required in the XML input document. The credential attributes are stored in a new XML element within the top level XML container `<Credential>`. For example:

```
<HTTPResponse>
  <Credential>
    <Attributes>
      <Attribute name=AZN_CRED_PRINCIPAL_NAME>testuser</Attribute>
    </Attributes>
  </Credential>
      …
</HTTPResponse>
```

Note: This configuration entry only applies to XSLT transformation rules and is ignored for Lua transformation rules.

Options

name
Name of the credential attribute.

Usage

This stanza entry is optional.

Default value

None.

Example

```
cred-attr-name = AZN_CRED_PRINCIPAL_NAME
```

lua-ldap-ca-cert-label

Use the `lua-ldap-ca-cert-label` stanza entry to define the label of the CA certificate that is used to verify the LDAP server certificate.

Syntax

```
lua-ldap-ca-cert-label = label
```

Description

This entry is used to specify the label of the certificate from the webseal-cert-keyfile key file that can be used to verify the server certificate presented by the LDAP server when you are using the lualdap Lua module.

Note: This configuration entry only applies to Lua transformation rules and is ignored for XSLT transformation rules.

Options

`label`
The label of the CA certificate.

Usage

This stanza entry is optional.

Default value

None.

Example

```
lua-ldap-ca-cert-label = my-ldap-server
```

lua-max-pool-size

Use the `lua-max-pool-size` stanza entry to define the maximum size of the pool of cached Lua scripting handles.

Syntax

```
lua-max-pool-size = entries
```

Description

This entry is used to define the maximum number of Lua scripting handles which will be cached in a pool for reuse. Reusing a Lua scripting handle will help to improve performance but will also have the impact of increasing memory usage by the WebSEAL process.

Note: This configuration entry only applies to Lua transformation rules and is ignored for XSLT transformation rules.

Options

`entries`
The maximum size of the pool of cached Lua scripting handles.

Usage

This entry is optional.

Default value

50

Example

```
lua-max-pool-size = 200
```

request-match

Use this entry to define the pattern to be matched against the HTTP request line, which includes method, URI, and protocol.

Syntax

```
request-match = {request|preazn|postazn|postauthn|response}:<request-line>
```

Description

This entry defines the pattern to be matched against the HTTP request line, which includes method, URI, and protocol.

You can also match a request by using a host header. Use this option to selectively enable this function for a particular virtual host junction. To selectively match an entry based on a particular host header, add a prefix to the *<request-line>* with the string `[<host>]`.

When you are defining the `request-match` entry, you also need to define the stage in the processing flow at which the rule is triggered. The following table lists the supported options:

Option	Description
`request`	The rule is triggered when the request is first received by WebSEAL. Credential attributes are not available at this stage in the processing flow.
`preazn`	The rule is triggered immediately before the standard authorization decision logic. It can be used to implement your own Lua based authorization decisions. This stage is only valid for Lua scripts.
`postazn`	The rule is triggered immediately after the authorization decision is made.
`postauthn`	The rule is only triggered after an authentication event. It can be used to add extended attributes to the credential. **Note:** The request-match is made against the name of the authentication mechanism rather than the HTTP request line. The name of the authentication mechanism is located within the *AZN_CRED_AUTH_METHOD* attribute of an authenticated credential.
`response`	The rule is triggered after the response has been received from the junction.

Options

[request |preazn|postazn|postauthn|response]
Determines the location in the processing flow at which the rule is to be triggered. This value can be **`request`**, **`preazn`**, **`postazn`**, **`postauthn`**, or **`response`**.

request-line
Contains the request line to be matched against. The pattern matching is case-sensitive. You can use wildcard characters "`*`" and "`?`".

Usage

This stanza entry is optional.

You can specify multiple entries if needed.

Default value

None.

Example

```
request-match = request:GET /index.html HTTP/1.1
request-match = postazn:GET /jct/* *
request-match = response:[www.ibm.com]GET /login/*
request-match = postauthn:password
```

xslt-buffer-size

Use the `xslt-buffer-size` stanza entry to define the maximum size of the output XML document produced by XSLT when you are evaluating the HTTP transformation rule.

Syntax

```
xslt-buffer-size = byte_limit
```

Description

The maximum size of the output XML document produced by XSLT when you are evaluating the HTTP transformation rule.

Note: This configuration entry only applies to XSLT transformation rules and is ignored for Lua transformation rules.

Options

byte_limit
The maximum limit of the output XML document that is produced by XSLT when the HTTP transformation rule is evaluated.

Usage

This entry is optional.

Default value

4096

Note: The value is ignored if it is set below the default value.

Example

```
xslt-buffer-size = 4096
```

[http-updates] stanza

Use the **[http-updates]** stanza to configure WebSEAL so that it can communicate with an HTTP server to retrieve updates to files.

update-url

Use the **update-url** stanza entry to define the URL that references the HTTP file.

Syntax

```
update-url = URL
```

Description

Use this stanza entry to specify the URL for the HTTP file.

Options

URL
The URL that contains the HTTP file.

Usage

This stanza entry is required.

Default value

None.

Example

```
update-url = https://99.n.example.com/74767/api/snippets
```

proxy

Use the **proxy** stanza entry to define the proxy server that is used when connecting to the HTTP server.

Syntax

```
proxy = host:port
```

Description

The proxy server that is used when connecting to the HTTP server.

Options

host*:*port
Specify the host name and port number of the proxy server.

Usage

This stanza entry is optional. If this entry is not present, then traffic goes to the server directly rather than going through a proxy.

Default value

None.

Example

```
proxy = www.example.com:8080
```

replace

Use the **replace** stanza entry to perform a search and replace operation on text that is contained in the updated files.

Syntax

```
replace = <search-patten>|<replace-text>
```

Description

Define the search pattern and the replacement text.

Options

<search-patten>
The regular expression pattern that is to be matched.

Note: The "|" character cannot be used in the search-pattern text.

<replace-text>
The text that will replace the matched text.

Usage

Multiple instances of this configuration entry can be used if multiple substitutions are required.

Default value

None.

Example

```
replace = .*old|new
```

ssl-keyfile-label

Use the **ssl-keyfile-label** stanza entry to define the label of the certificate that is used for authentication to the HTTP server. This stanza entry is required only if client certificate authentication is required by the update server.

Syntax

```
ssl-keyfile-label = label_name
```

Description

This certificate must be present in the certificate database that is used for junction communication.

Options

label_name
The label of the certificate that is used for authentication to the HTTP server.

Usage

This stanza entry is optional.

Default value

None.

Example

```
ssl-keyfile-label = "samplelabel"
```

ssl-server-dn

Use the **ssl-server-dn** stanza entry to define the DN of the server. Use this configuration if you want to validate the DN that is contained in the server certificate of the update server.

Syntax

```
ssl-server-dn = DN_value
```

Description

This configuration entry is used only if an SSL connection is established with the server and an SSL key file label has been specified.

Options

DN_value
The DN of the server

Usage

This stanza entry is optional.

Default value

None.

Example

```
ssl-server-dn = CN=Verify Access,OU=SecureWay,O=Tivoli,C=US
```

poll-period

Use the **poll-period** stanza entry to define the frequency that the update server is polled for updates.

Syntax

```
poll-period = seconds
```

Description

This value is defined in seconds.

Options

seconds
The frequency in seconds that the update server is polled for updates.

Usage

This stanza entry is required.

Default value

None.

Example

```
poll-period = 3600
```

[ICAP:<resource>] stanza

Use the **[ICAP:<resource>]** stanza to define a single ICAP resource.

The <resource> component of the stanza name must be changed to the actual name of the resource. To enable the ICAP resource for a particular object, a POP must be attached to the appropriate part of the object space. This POP must contain an extended attribute with the name ICAP, and a value that is equal to the name of the configured ICAP resource.

URL

Use the **URL** stanza entry to define the complete URL on which the ICAP server is expecting requests and whether to use TCP or SSL for the connection to the server.

Syntax

```
URL = URL_string
```

Description

The complete URL on which the ICAP server is expecting requests.

Options

URL_string
URL string.

- To establish a TCP connection to the ICAP server, use the following format:

  ```
  URL = icap://<ICAP Server host/IP[:port]>/<path>
  ```

- To establish an SSL connection to the ICAP server, use the following format:

  ```
  URL = icaps://<ICAP Server host/IP[:port]>/<path>
  ```

  The system uses the keystore that is configured in the **[junction]** stanza if it exists. If not, the system uses the keystore that is configured in the **[ssl]** stanza.

Usage

Required

Default value

None

Example

```
URL = icap://icap.example.net:1344/filter?mode=strict
```

Note: In this example, a TCP connection is established.

```
URL = icaps://icap.example.net:1345/filter?mode=strict
```

Note: In this example, an SSL connection is established.

transaction

Use the **transaction** stanza entry to define the transaction for which the resource is invoked.

Syntax

```
transaction = {req | rsp}
```

Description

The transaction for which the resource is invoked.

Options

req
The ICAP server is invoked on the HTTP request.

rsp
The ICAP server is invoked on the HTTP response.

Usage

Required

Default value

None

Example

```
transaction = req
```

timeout

Use the **timeout** stanza entry to define the maximum length of time that WebSEAL waits for a response from the ICAP server.

Syntax

```
timeout = seconds
```

Description

The maximum length of time (in seconds) that WebSEAL waits for a response from the ICAP server.

Options

seconds
The time in seconds, that WebSEAL waits for a response from the ICAP server.

Usage

Required

Default value

None

Example

```
timeout = 120
```

ssl-keyfile-label

If client certificate authentication is required for the SSL connection to the ICAP server, use this entry to define the label of the certificate to use from the keystore.

Syntax

```
ssl-keyfile-label = label
```

Description

This entry identifies the label of the certificate in the keystore to use when the reverse proxy establishes an SSL connection to the ICAP server.

Options

label
The label of the client certificate to use from the keystore.

Usage

This entry is required only if client certificate authentication is needed.

Default value

None

Example

```
ssl-keyfile-label = my_certificate
```

[interfaces] stanza

interface_name

User the `interface_name` to define additional interfaces on which this WebSEAL instance can receive requests.

Syntax

```
interface_name = property=value[;property=value...]
```

Description

This stanza is used to define additional interfaces on which this WebSEAL instance can receive requests.

A network interface is defined as the combined set of values for a specific group of properties that include HTTP or HTTPS port setting, IP address, worker threads setting, and certificate handling setting.

Options

property

Interface property. Can be selected from:

```
network-interface=<ipAddress>
http-port=<port> | disabled
https-port=<port> | disabled
web-http-port=<port> | disabled
web-http-protocol=http |https
certificate-label=<keyFileLabel>
accept-client-certs=never | required | optional | prompt_as_needed
worker-threads=<count> | default
secondary-port= <port> | disabled
always-neg-tls= yes | no
enable-http2=yes | no
http2-max-connections=<number_of_connections>
http2-header-table-size=<header_size>
http2-max-concurrent-streams=<number_of_stream>
http2-initial-window-size=<window_size>
http2-max-frame-size=<frame_size>
http2-max-header-list-size=<list_size>
http-proxy-protocol= yes | no
https-proxy-protocol= yes | no
websocket-max-worker-threads = <count>
```

value

Value of the property. Default values, if not present, include:

```
network-interface=0.0.0.0
http-port=disabled
https-port=disabled
enable-http2=no
certificate-label= (Uses key marked as default in key file.)
accept-client-certs=never
worker-threads=default
http-proxy-protocol=no
https-proxy-protocol=no
websocket-max-worker-threads = (Uses the global pool of WebSocket worker threads)
```

Usage

Entries in this stanza are optional.

Default value

None.

Example

(Entered as one line:)

```
support = network-interface=9.0.0.8;https-port=444;certificate-label=WS6;
worker-threads=16
```

[itim] stanza

This stanza contains the configuration options for the IBM Security Identity Manager Password Synchronization Plug-in. The Password Synchronization Plug-in synchronizes user passwords from IBM Security Verify Access for Web to IBM Security Identity Manager, previously known as IBM Tivoli Identity Manager.

For more information about this plug-in, see the *Password Synchronization Plug-in for IBM Security Verify Access Installation and Configuration Guide*, which you can find in the IBM Security Identity Manager Knowledge Center.

is-enabled

Syntax

```
is-enabled = {true|false}
```

Description

Determines whether the Password Synchronization Plug-in for IBM Security Identity Manager, is enabled.

Options

`true`
Enables the Password Synchronization Plug-in.

`false`
Disables the Password Synchronization Plug-in.

Usage

This stanza entry is optional.

Default value

`false`

Example

```
is-enabled = false
```

itim-server-name

Syntax

```
itim-server-name = <itim_server>
```

Description

Specifies the host name or IP address of the server that is running IBM Security Identity Manager.

Note: In a WebSphere® Application Server cluster environment, you must configure SSL for the IBM® HTTP Server. In a WebSphere Application Server single-server environment, you do not need to configure SSL for the IBM HTTP Server.

Options

<itim_server>
Specifies the host name or IP address of the IBM Security Identity Manager server that communicates with IBM Security Verify Access for Web.

Usage

This stanza entry is required when the **`is_enabled`** configuration entry in the **`[itim]`** stanza is set to `true`.

Default value

None.

Example

```
itim-server-name = identityMgr01.ibm.com
```

itim-servlet-context

Syntax

```
itim-servlet-context = <directory_path>
```

Description

Indicates the password synchronization context root on the application server.

Options

<directory_path>
Specifies the directory path for the password synchronization context root on the application server.

Usage

This stanza entry is required when the **is_enabled** configuration entry in the **[itim]** stanza is set to `true`.

Default value

`/passwordsynch/synch`.

Example

```
itim-servlet-context = /passwordsynch/synch
```

keydatabase-file

Syntax

```
keydatabase-file = <file_name>
```

Description

Specifies the name of the key database file.

Options

<file_name>
The name of the key database file.

Usage

This stanza entry is required when the **is_enabled** configuration entry in the **[itim]** stanza is set to `true`.

Default value

None.

Example

```
keydatabase-file = revpwdsync.kdb
```

keydatabase-password

Syntax

```
keydatabase-password = <db_password>
```

Description

Specifies the password for the key database in the **keydatabase-file**.

Note: The IBM Security Verify Access appliance uses `stash` files to manage the passwords for key files. As a result, key file passwords are not available to the administrator of the appliance.

If you do not know the password for the key database file, you can use the **keydatabase-password-file** entry to specify the name of the password `stash` file instead. If you configure the **keydatabase-password-file** entry, you can leave the **keydatabase-password** entry unconfigured.

The Password Synchronization Plug-in requires knowledge of the database password. Therefore, if you do not configure the **keydatabase-password-file** entry, you must configure the **keydatabase-password** entry. To complete this configuration, follow this process:

1. Create the key file externally to the appliance. Use a known password to generate the new key file.
2. Import the key file on to the appliance.
3. Configure the **keydatabase-password** configuration entry with the known password for the Password Synchronization Plug-in.

Options

<db_password>
Specifies the password for the key database file.

Usage

If the **is_enabled** configuration entry in the **[itim]** stanza is set to `true`, you must set one of the following entries for the key database password:

- **keydatabase-password**
- **keydatabase-password-file**

Note: If there is a value configured for both of these entries, WebSEAL uses the **keydatabase-password**.

Default value

None.

Example

```
keydatabase-password = myPassword1
```

keydatabase-password-file

Syntax

```
keydatabase-password-file = <password_stash_file>
```

Description

Specifies the name of the `stash` file that stores the password for the key database.

Options

<password_stash_file>
Specifies the name of the `stash` file that stores the password for the key database.

Usage

If the **is_enabled** configuration entry in the **[itim]** stanza is set to `true`, you must set one of the following entries for the key database password:

- **keydatabase-password**
- **keydatabase-password-file**

Note: If there is a value configured for both of these entries, WebSEAL uses the **keydatabase-password**.

Default value

None.

Example

```
keydatabase-password-file = dbPassword.sth
```

principal-name

Syntax

```
principal-name = <user_name>
```

Description

Specifies an IBM Security Identity Manager user ID that has the necessary permissions to complete the **check** and **synchronization** operations.

Note: Do not use the **ITIM manager** account for this purpose. Create a separate account on the IBM Security Identity Manager server with the same permissions.

Options

<user_name>
Specifies the name of the IBM Security Identity Manager user that the Password Synchronization Plug-in can use to request synchronization operations.

Usage

This stanza entry is required when the **is_enabled** configuration entry in the **[itim]** stanza is set to `true`.

Default value

None.

Example

```
principal-name = admin_userA
```

principal-password

Syntax

```
principal-password = <user_password>
```

Description

Specifies the password of the IBM Security Identity Manager user that is specified by **principal-name**.

Options

<user_password>
Specifies the password for the IBM Security Identity Manager account.

Usage

This stanza entry is required when the **is_enabled** configuration entry in the **[itim]** stanza is set to `true`.

Default value

None.

Example

```
principal-password = myPassword1
```

service-password-dn

Syntax

```
service-password-dn = <service_pseudo_dn>
```

Description

Defines the pseudo–distinguished name of the service that issues the password synchronization request.

The Password Synchronization Plug-in uses the **service-password-dn** pseudo-distinguished name for requests that use the standard password authentication method. If this configuration entry is specified, it overrides **service-source-dn** when using the password authentication method.

Note: You can specify more than one pseudo-distinguished name. Separate the pseudo-distinguished names with a semicolon (`;`) character. The Password Synchronization Plug-in iterates through the list of service names until it finds an account for one of the services. If the Password Synchronization Plug-in cannot find an account for the specified services, it returns an error message.

Each pseudo-distinguished name is a comma-separated list of the following attributes:

- The **erservicename** attribute of the Security Verify Access service name, as defined in IBM Security Identity Manager. For example, `erservicename=TAM 6.0 Service`.
- The **o** attribute of the organization to which the service belongs. For example, `o=International Business Machines`.
- The **ou** and **dc** attributes from the service distinguished name in IBM Security Identity Manager. For example, `ou=IBM,dc=com`.

The pseudo-distinguished name that is formed from these example values is: `erservicename=TAM 6.0 Service,o=International Business Machines, ou=IBM,dc=com`.

Options

<service_pseudo_dn>
Specifies the service pseudo–distinguished name for the standard password authentication method.

Usage

If the **is_enabled** configuration entry in the **[itim]** stanza is set to `true`, then you must configure at least one of the following configuration entries:

- **service-source-dn**
- **service-password-dn**
- **service-token-card-dn**

Default value

None.

Example

```
service-password-dn = erservicename=ISVA Employees Service,o=IBM,ou=IBM,dc=com
```

service-source-dn

Syntax

```
service-source-dn = <service_pseudo_dn>
```

Description

Defines the pseudo–distinguished name of the service that issues the password synchronization request. The **service-source-dn** is for the pseudo-distinguished name for all authentication methods.

Note: You can specify more than one pseudo-distinguished name in the value of this configuration entry. Separate the pseudo-distinguished names with a semicolon (`;`) character. The Password Synchronization Plug-in iterates through the list of service names until it finds an account for one of the services. If the Password Synchronization Plug-in cannot find an account for the specified services, it returns an error message.

Each pseudo-distinguished name is a comma-separated list of the following attributes:

- The **erservicename** attribute of the Security Verify Access service name, as defined in IBM Security Identity Manager. For example, `erservicename=TAM 6.0 Service`.
- The **o** attribute of the organization to which the service belongs. For example, `o=International Business Machines`.
- The **ou** and **dc** attributes from the service distinguished name in IBM Security Identity Manager. For example, `ou=IBM,dc=com`.

The pseudo-distinguished name that is formed from these example values is: `erservicename=TAM 6.0 Service,o=International Business Machines, ou=IBM,dc=com`.

Options

<service_pseudo_dn>
Specifies the service pseudo–distinguished name for all authentication methods.

Usage

If the **is_enabled** configuration entry in the **[itim]** stanza is set to `true`, then you must configure at least one of the following configuration entries:

- **service-source-dn**
- **service-password-dn**
- **service-token-card-dn**

Default value

None.

Example

```
service-source-dn = erservicename=ISVA Employees Service,o=IBM,ou=IBM,
dc=com;erservicename=TAM Customers Service,o=IBM,ou=IBM,dc=com
```

service-token-card-dn

Syntax

```
service-token-card-dn = <service_pseudo_dn>
```

Description

Defines the pseudo–distinguished name of the service that issues the password synchronization request.

The Password Synchronization Plug-in uses the **service-token-card-dn** pseudo-distinguished name for requests that use the token card authentication method. If this configuration entry is specified, it overrides **service-source-dn** when using the token card authentication method.

Note: You can specify more than one pseudo-distinguished name. Separate the pseudo-distinguished names with a semicolon (`;`). The Password Synchronization Plug-in iterates through the list of service names until it finds an account for one of the services. If the Password Synchronization Plug-in cannot find an account for the specified services, it returns an error message.

Each pseudo-distinguished name is a comma-separated list of the following attributes:

- The **erservicename** attribute of the Security Verify Access service name, as defined in IBM Security Identity Manager. For example, `erservicename=TAM 6.0 Service`.
- The **o** attribute of the organization to which the service belongs. For example, `o=International Business Machines`.

- The **ou** and **dc** attributes from the service distinguished name in IBM Security Identity Manager. For example, `ou=IBM,dc=com`.

The pseudo-distinguished name that is formed from these example values is: `erservicename=TAM 6.0 Service,o=International Business Machines, ou=IBM,dc=com`.

Options

<service_pseudo_dn>
Specifies the service pseudo–distinguished name for the token card authentication method.

Usage

If the **is_enabled** configuration entry in the **[itim]** stanza is set to `true` then you must configure at least one of the following configuration entries:

- **service-source-dn**
- **service-password-dn**
- **service-token-card-dn**

Default value

None.

Example

```
service-token-card-dn = erservicename=ISVA Employees Service,o=IBM,ou=IBM,dc=com
```

servlet-port

Syntax

```
servlet-port = <port_number>
```

Description

Specifies the port number for communicating with the IBM Security Identity Manager server that is specified by the **itim-server-name** configuration entry.

The default HTTPS port is 9443 for a single server configuration and 443 for a IBM Security Identity Manager cluster with HTTP SSL configured.

Options

<port_number>
Specifies the port number for communication with the IBM Security Identity Manager server.

Usage

This stanza entry is required when the **is_enabled** configuration entry in the **[itim]** stanza is set to `true`.

Default value

9443

Example

```
servlet-port = 9443
```

[jdb-cmd:replace] stanza

jct-id=search-attr-value|replace-attr-value

Syntax

```
jct-id=search-attr-value|replace-attr-value
```

Description

Defines the mapping rules for the **jdb import** command. These mapping rules are applied to each attribute in the junction archive file before you import the new junction database.

Options

jct-id
Refers to the junction point for a standard junction which includes the leading '/' (slash) or the virtual host label for a virtual host junction.

search-attr-value
Specifies the attribute value in the junction definition for which you want to search and replace.

replace-attr-value
Specifies the new attribute value in the junction definition for which you want to search and replace.

Usage

This stanza entry is not required.

Default value

None.

Example

```
/test-jct = webseal.au.ibm.com|webseal.gc.au.ibm.com
```

[junction] stanza

allow-backend-domain-cookies

Use the **allow-backend-domain-cookies** stanza entry to control whether WebSEAL sends domain cookies from a back-end server to a client.

Syntax

```
allow-backend-domain-cookies = {yes|no}
```

Description

Indicates whether WebSEAL can send domain cookies from a back-end server to a client.

You can customize this configuration item for a particular junction by adding the adjusted configuration item to a **[junction:*{junction_name}*]** stanza.

where *{junction_name}* refers to the junction point for a standard junction (including the leading / character) or the virtual host label for a virtual host junction.

Options

yes
WebSEAL is able to send domain cookies from a back-end server to a client.

no
WebSEAL is not able to send domain cookies from a back-end server to a client.

Usage

This stanza entry is required.

Default value

no

Example

```
allow-backend-domain-cookies = no
```

always-send-kerberos-tokens

Indicates whether WebSEAL sends a security token for every HTTP request or whether WebSEAL waits for a 401 response before it adds the security token.

Syntax

```
always-send-kerberos-tokens = {yes|true|no|false}
```

Description

You can customize this configuration item for a particular junction by adding the adjusted configuration item to a **[junction:{junction_name}]** stanza.

Options

yes
Enable.

true
Enable.

no
Disable.

false
Disable.

Usage

This stanza entry is required if Kerberos SSO authentication for junctions is enabled.

Default value

no.

Example

```
always-send-kerberos-tokens = no
```

basicauth-dummy-passwd

Use the **basicauth-dummy-passwd** stanza entry to specify the global password for WebSEAL to use when it supplies basic authentication data over junctions that were created with the `-b supply` argument.

Syntax

```
basicauth-dummy-passwd = dummy_password
```

Description

Global password that WebSEAL uses when it is supplying basic authentication data over junctions that were created with the `-b supply` argument.

Options

dummy_password
Global password that WebSEAL uses when it is supplying basic authentication data over junctions that were created with the `-b supply` argument. Passwords must consist of ASCII characters.

Usage

This stanza entry is required.

Default value

dummy

Example

```
basicauth-dummy-passwd = dummy
```

connect-timeout

Syntax

```
connect-timeout = number_of_seconds
```

Description

Integer value indicating the timeout, in seconds, for establishing a connection to a junctioned server.

You can customize this configuration item for a particular junction by adding the adjusted configuration item to a **[junction:*{junction_name}*]** stanza.

where *{junction_name}* refers to the junction point for a standard junction (including the leading / character) or the virtual host label for a virtual host junction.

Options

number_of_seconds
Integer value indicating the timeout, in seconds, for establishing a connection to a junctioned server. The minimum value is 0. When the value is 0, there is no timeout. WebSEAL does not impose a maximum value.

Usage

This stanza entry is required.

Default value

30

```
connect-timeout = 30
```

crl-ldap-server

Use the **crl-ldap-server** stanza entry in the **[junction]** stanza to specify the LDAP server that WebSEAL can contact for CRL checking during authentication across SSL junctions.

Syntax

```
crl-ldap-server = server_name
```

Description

Specifies the Server to be contacted to obtain Certificate Revocation Lists (CRL).

Options

server_name
This parameter can be set to one of two types of values:

1. The name of the LDAP server to be referenced as a source for Certificate Revocation Lists (CRL) during authentication across SSL junctions. If this is used, you may also need to set the following parameters:
   - crl-ldap-server-port
   - crl-ldap-user
   - crl-ldap-user-password
2. The literal string “URI”. In the case where no direct LDAP Server is available, this allows GSKit to obtain revocation information from LDAP or the HTTP Servers as specified by the CA in the CRL Distribution Point (CDP) extension of the certificate.

Usage

This stanza entry is optional.

Default value

None.

Example

```
crl-ldap-server = diamond.example.com
```

crl-ldap-server-port

Use the **crl-ldap-server-port** entry in the **[junction]** stanza to set the port number for WebSEAL to use when it communicates with the LDAP server specified in **crl-ldap-server**.

Syntax

```
crl-ldap-server-port = port_number
```

Description

Port number for communication with the LDAP server specified in **crl-ldap-server**. The LDAP server is referenced for Certificate Revocation List (CRL) checking during authentication across SSL junctions.

Options

port_number
Port number for communication with the LDAP server specified in **crl-ldap-server**.

Usage

This stanza entry is optional. When **crl-ldap-server** is specified, this stanza entry is required.

Default value

None.

Example

```
crl-ldap-server-port = 389
```

crl-ldap-user

Use the **crl-ldap-user** entry in the **[junction]** stanza to specify an LDAP user who has permissions to retrieve the CRL on the LDAP server that is specified in **crl-ldap-server**.

Syntax

```
crl-ldap-user = user_DN
```

Description

Fully qualified distinguished name (DN) of an LDAP user who has permissions to retrieve the Certificate Revocation List.

Options

user_DN
Fully qualified distinguished name (DN) of an LDAP user who has permissions to retrieve the Certificate Revocation List. A null value for **crl-ldap-server** indicates that the SSL authenticator should bind to the LDAP server anonymously.

Usage

This stanza entry is optional.

Default value

None.

Example

```
crl-ldap-user = user_DN
```

crl-ldap-user-password

Use the **crl-ldap-user-password** entry in the **[junction]** stanza to provide the password for the LDAP user that is specified in **crl-ldap-user**.

Syntax

```
crl-ldap-user-password = password
```

Description

The password for the LDAP user specified in the **crl-ldap-user** stanza entry.

Options

password
The password for the LDAP user specified in the **crl-ldap-user** stanza entry.

Usage

This stanza entry is optional. When **crl-ldap-user** is specified, this stanza entry is required.

Default value

None.

Example

```
crl-ldap-user-password = mypassw0rd
```

disable-local-junctions

Use the **disable-local-junctions** stanza entry to control whether WebSEAL serves pages from a local web server through local junctions.

Syntax

```
disable-local-junctions = {yes|no}
```

Description

If local junctions are not used, you can disable the functionality with the `disable-local-junctions` configuration item.

Options

yes
Disables local junction functionality.

no
Enables local junction functionality.

Usage

Optional.

The following example enables local junction functionality:

```
disable-local-junctions=no
```

disable-on-ping-failure

Use the **disable-on-ping-failure** stanza entry to configure the Web Reverse Proxy to return an error when HTTP requests are received for junctioned servers which are currently failing the 'ping' operation.

Syntax

```
disable-on-ping-failure = {yes|no}
```

Description

If a ping to a junction fails, WebSEAL continues to send requests from clients to the junction. If this configuration entry is set to `true`, WebSEAL stops sending requests from clients to the junction while the ping operation is failing.

Options

yes
If set to `yes`, requests from a client are not sent to the junctioned server while the ping operation is failing.

no
If set to `no`, requests from a client are sent to the junctioned server while the ping operation is failing.

Usage

This stanza entry is optional.

Default value

no

Example

```
disable-on-ping-failure = no
```

disable-ssl-v2

Use the **disable-ssl-v2** entry in the **[junction]** stanza to control whether WebSEAL supports SSL version 2 for junction connections.

Syntax

```
disable-ssl-v2 = {yes|no}
```

Description

Disables support for SSL Version 2 for junction connections. Support for SSL v2 is disabled by default.

This configuration item may be customized for a particular junction by adding the adjusted configuration item to a `[junction:{junction_name}]` stanza, where *{junction_name}* refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

Options

yes
The value `yes` means support is disabled.

no
The value no means the support is enabled.

Usage

This stanza entry is optional. When not specified, the default is `yes`. The WebSEAL configuration sets this value.

Default value

`yes`

Example

```
disable-ssl-v2 = yes
```

disable-ssl-v3

Use the **disable-ssl-v3** entry in the **[junction]** stanza to control whether WebSEAL supports SSL version 3 for junction connections.

Syntax

```
disable-ssl-v3 = {yes|no}
```

Description

Disables support for SSL Version 3 for junction connections. Support for SSL V3 is enabled by default.

This configuration item may be customized for a particular junction by adding the adjusted configuration item to a `[junction:{junction_name}]` stanza, where *{junction_name}* refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

Options

yes
The value `yes` means support is disabled.

no
The value no means the support is enabled

Usage

This stanza entry is optional. When not specified, the default is no. The WebSEAL configuration sets this value.

Default value

no

Example

```
disable-ssl-v3 = no
```

disable-tls-v1

Use the **disable-tls-v1** entry in the **[junction]** stanza to control whether WebSEAL supports Transport Layer Security (TLS) version 1 for junction connections.

Syntax

```
disable-tls-v1 = {yes|no}
```

Description

Disables support for TLS Version 1 for junction connections. Support for TLS V1 is enabled by default.

This configuration item may be customized for a particular junction by adding the adjusted configuration item to a `[junction:{junction_name}]` stanza, where *{junction_name}* refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

Options

yes
The value `yes` means support is disabled.

no
The value no means the support is enabled.

Usage

This stanza entry is optional. When not specified, the default is no. The WebSEAL configuration sets this value.

Default value

no

Example

```
disable-tls-v1 = no
```

disable-tls-v11

Use the **disable-tls-v11** entry in the **[junction]** stanza to control whether WebSEAL supports Transport Layer Security (TLS) version 1.1 for junction connections.

Syntax

```
disable-tls-v11 = {yes|no}
```

Description

Determines whether WebSEAL supports TLS version 1.1 for junction connections. Support for TLS v1.1 is enabled by default.

This configuration item may be customized for a particular junction by adding the adjusted configuration item to a `[junction:{junction_name}]` stanza, where *{junction_name}* refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

Options

yes
The value `yes` disables support for TLS version 1.1.

no
The value `no` enables support for TLS version 1.1.

Usage

This stanza entry is optional. If this entry is not specified, the default is `no`.

Default value

no

Example

```
disable-tls-v11 = no
```

disable-tls-v12

Use the **disable-tls-v12** entry in the **[junction]** stanza to control whether WebSEAL supports Transport Layer Security (TLS) version 1.2 for junction connections.

Syntax

```
disable-tls-v12 = {yes|no}
```

Description

Determines whether WebSEAL supports Transport Layer Security (TLS) version 1.2 for junction connections. Support for TLS v1.2 is enabled by default.

This configuration item may be customized for a particular junction by adding the adjusted configuration item to a `[junction:{junction_name}]` stanza, where *{junction_name}* refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

Options

yes
The value `yes` disables support for TLS version 1.2.

no
The value `no` enables support for TLS version 1.2.

Usage

This stanza entry is optional. If this entry is not specified, the default is `no`.

Default value

no

Example

```
disable-tls-v12 = no
```

disable-tls-v13

Use the `disable-tls-v13` entry in the `[junction]` stanza to control whether support for TLS version 1.3 is enabled in WebSEAL.

Syntax

`disable-tls-v13 = {yes|no}`

Description

Determines whether WebSEAL supports Transport Layer Security (TLS) version 1.3 for junction connections. Support for TLS version 1.3 is disabled by default.

This configuration item may be customized for a particular junction by adding the adjusted configuration item to a `[junction:`*`{junction_name}`*`]` stanza, where *`{junction_name}`* refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

Options

yes
Disables support for TLS version 1.3

no
Enables support for TLS version 1.3

Usage

This stanza entry is optional. If this entry is not specified, the default is `yes`.

disable-tls-v13=no disables DPWNS0301W messages.

Default value

`yes`

Example

`disable-tls-v13 = no`

dont-reprocess-jct-404s

Use the **`dont-reprocess-jct-404s`** stanza entry to control whether WebSEAL reprocesses requests that fail with an HTTP 404 error by prepending the junction name to the URL.

Syntax

```
dont-reprocess-jct-404s = {yes|no}
```

Description

If a resource cannot be found on a back-end server, that server returns an HTTP 404 error. The **dont-reprocess-jct-404s** stanza entry controls whether or not WebSEAL processes the request again by prepending the junction name to the URL.

You should never need to enable this stanza entry if you follow this best practice for junctions: **The junction name should not match any directory name used in the Web space of the back-end server if HTML pages from that server contain programs (such as JavaScript or applets) with server-relative URLs to that directory.**

The following scenario can occur when one does not adhere to this best practice for junctions:

1. A resource is located in the following subdirectory (using the same name as the junction) on the back-end server: `/jct/page.html`.
2. A page received by the client from this back-end server contains the following URL: `/jct/page.html`.
3. When the link is followed, WebSEAL can immediately process the request because it recognizes what it thinks is the junction name in the URL. No configured URL modification technique is required.
4. At the time the request is forwarded to the back-end server, the junction name (`/jct`) removed from the URL. The resource (`/page.html`) is not found at the root of the back-end server file system. The server returns a 404 error.
5. If WebSEAL is configured for `dont-reprocess-jct-404s=no`, it reprocesses the URL and prepends the junction name to the original URL: `/jct/jct/page.html`.
6. Now the resource is successfully located at `/jct/page.html` on the back-end server.

Notes:

- The default behavior in WebSEAL is to reprocess a request URL after an HTTP 404 error is returned from the back-end server. You can set the value of **dont-reprocess-jct-404s** to `yes` to override this default behavior.
- If the **reprocess-root-jct-404s** entry (also in the **[junction]** stanza) has been set to `yes` then root junction resource requests that result in a HTTP 404 error *will* be reprocessed regardless of the setting of this **dont-reprocess-jct-404s** stanza entry.

Options

yes
When the back-end server returns an HTTP 404 error, do not reprocess the request URL.

no
When the back-end server returns an HTTP 404 error, reprocess the request URL by prepending the junction name to the existing URL.

Usage

This stanza entry is required.

Default value

The default value in the template configuration file is `yes`.

Example

```
dont-reprocess-jct-404s = yes
```

dynamic-addresses

Use the **dynamic-addresses** stanza entry to control whether the junction server host name is resolved to its IP address immediately before every communication with the junction server.

Syntax

```
dynamic-addresses = {yes|no}
```

Description

Indicates when the junction server host name is resolved to its corresponding IP address and used in communication with the junction server.

You can customize this configuration item for a particular junction by adding the adjusted configuration item to a **[junction:*{junction_name}*]** stanza.

where *{junction_name}* refers to the junction point for a standard junction (including the leading / character) or the virtual host label for a virtual host junction.

Options

yes
The junction server host name is resolved to its corresponding IP address immediately before any communication with the junction server.

If this configuration entry is set to yes, you can use the **dynamic-addresses-ttl** configuration entry to specify the length of time (in seconds) that a resolved IP address will be cached before it is discarded and another name resolution is attempted (time-to-live).

no
The junction server host name is resolved to its corresponding IP address and this address is used for subsequent communication with the junction server.

Usage

This stanza entry is required.

Default value

no

Example

```
dynamic-addresses = no
```

dynamic-addresses-ttl

Use the **dynamic-addresses-ttl** stanza entry to specify the length of time (in seconds) that a resolved IP address will be cached before it is discarded and another name resolution is attempted (time-to-live).

Syntax

```
dynamic-addresses-ttl = seconds
```

Description

If the **dynamic-addresses** configuration entry is set to yes, this configuration entry specifies the length of time (in seconds) that a resolved IP address will be cached before it is discarded and another name resolution is attempted (time-to-live).

You can customize this configuration item for a particular junction by adding the adjusted configuration item to a **[junction:*{junction_name}*]** stanza, where *{junction_name}* refers to the junction point for a standard junction (including the leading / character) or the virtual host label for a virtual host junction.

Options

seconds
The length of time (in seconds) that a resolved IP address will be cached before it is discarded and another name resolution is attempted (time-to-live).

Usage

This stanza entry is optional.

Default value

None

Example

```
dynamic-addresses-ttl = 0
```

expect-hdr-timeout

Use this entry to set a timeout value for requests which contain the ‘expect: 100-continue’ header.

Syntax

```
expect-hdr-timeout = <timeout>
```

Description

The length of time, in seconds, that WebSEAL will wait for the initial ‘100 continue’ or ‘417 expectation failed’ status response from a junctioned server after having sent a request which contains the ‘expect: 100-continue’ header.

Options

<timeout>

The length of time, in seconds, to wait for a response to an ‘expect: 100-continue’ request.

Usage

This stanza entry is optional.

Default value

5

Example

```
expect-hdr-timeout = 5
```

failover-on-read

Use this entry to specify whether to retry requests to replicated junction servers or junction servers that are configured to use persistent connections when an error occurs on the initial request.

Syntax

```
failover-on-read = {yes | no}
```

Description

There might be some environments where it is undesirable to retry requests to replicated junction servers or junction servers that are configured to use persistent connections when an error occurs on the initial request.

The behavior can be changed so that a retry request is not made by setting this option in the **[junction]** stanza.

Options

yes
Retry requests when an error occurs on the initial request.

no
Do not retry requests when an error occurs on the initial request.

Usage

This stanza entry is optional.

Default value

`yes`

Example

```
failover-on-read = no
```

flush-cookie

Use the `flush-cookie` stanza entry to specify the browser cookies which should be cleared when a session is first established.

Syntax

```
flush-cookie = <cookie-details>
```

Description

Specifies a cookie which will be cleared in the browser when a session, either authenticated or unauthenticated, is first established. This provides a mechanism to reset stale cookies which might be present in the browser. If you are really concerned about sensitive cookies being left in the browser the embedded WebSEAL cookie jar functionality should be configured to internally store those sensitive cookies.

Multiple attributes can be added to the cookie definition, delimited by the ';' character. The supported attributes include: Path, Domain. The constructed cookie (which can include the path and domain attributes) must exactly match the cookie which is to be flushed, otherwise the browser will not be able to locate and clear the correct cookie.

This entry may be repeated multiple times, once for each cookie which is to be flushed.

Options

<cookie-details>
The name of the cookie which is to be deleted, along with optional path and domain attributes.

Usage

This stanza entry is optional.

Default value

None.

Example

```
flush-cookie = MyAppCookie
flush-cookie = MyJctCookie;Path=/jct/
flush-cookie = MyDomainCookie;Path=/jct/;Domain=ibm.com
```

persistent-failover-on-read

When persistent connections are enabled, this entry specifies whether retries on error conditions will also be made to the same server on a different connection if a request on a particular connection fails.

Syntax

```
persistent-failover-on-read = {yes | no}
```

Description

When persistent connections are enabled, retries on error conditions will also be made to the same server on a different connection if a request on a particular connection fails. This behavior can be changed so that these retry requests are not made by setting this option in the **[junction]** stanza.

Options

yes
Retry requests when an error occurs on the initial request.

no
Do not retry requests when an error occurs on the initial request.

Usage

This stanza entry is optional.

Default value

yes

Example

```
persistent-failover-on-read = no
```

gso-credential-learning

Use this entry to enable or disable the learning capability for GSO junctions.

Syntax

```
gso-credential-learning = {yes|no|true|false}
```

Description

This stanza entry controls whether the learning capability is enabled for GSO junctions.

If the learning ability is enabled and existing credential information is not available for the user, the BA prompt is returned to the user. The credential information for the user is then stored for future use on a subsequent successful authentication. An authentication is deemed to be successful if the junctioned web server does not return a 4xx or 5xx response.

Options

yes/true
Enable the learning ability for GSO junctions.

no/false
Disable the learning ability for GSO junctions.

Usage

This stanza entry is optional.

Default value

no

Example

```
gso-credential-learning = no
```

gso-obfuscation-key

Use this stanza entry to set the key for obfuscating any passwords that are managed by the GSO RESTful web service.

Syntax

```
gso-obfuscation-key = key
```

Description

If you want to obfuscate the passwords that are managed by the GSO RESTful web service, set this stanza entry. The passwords are obfuscated by performing an AES-CBC encryption on the data by using the supplied key. The appliance provides a web service to produce the obfuscated GSO password. For more information about this web service, see REST API documentation.

Note: The passwords are obfuscated by AES-128-CBC encryption, which is AES cipher-block-chaining encryption with a 128 bit key. If the supplied key is less than 16 bytes in length, the key will be right-padded with one or more "0" characters to bring the key up to 16 bytes in length.

If this stanza entry is not set, the passwords that are managed by the GSO RESTful web service are not obfuscated.

Options

key
The key for obfuscating the passwords that are managed by the GSO RESTful web service.

Usage

This stanza entry is optional.

Default value

None

Example

```
gso-obfuscation-key = jkhdc879e$*&^jcw98y
```

http2-header-table-size

Use the **http2-header-table-size** stanza entry to define the max header table size for an HTTP/2 network connection.

Syntax

```
http2-header-table-size = table_size
```

Description

This stanza entry defines the maximum size in bytes that WebSEAL accepts for header compression table (RFC 7541). There is one table per HTTP/2 network connection to a junctioned server.

This configuration item may be customized for a particular junction by adding the adjusted configuration item to a `[junction:{jct_id}]` stanza, where `'{jct-id}'` refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

Options

table_size
The maximum size in bytes that WebSEAL will accept for header compression table.

Usage

This stanza entry is optional.

Default value

None.

Example

```
http2-header-table-size = 4096
```

http2-initial-window-size

Use the **http2-initial-window-size** stanza entry to define the maximum number of unacknowledged bytes WebSEAL can accept per active multiplexed stream.

Syntax

```
http2-initial-window-size = number_of_bytes
```

Description

This stanza entry defines the maximum number of unacknowledged bytes WebSEAL can accept per active multiplexed stream to a junctioned server.

This configuration item may be customized for a particular junction by adding the adjusted configuration item to a `[junction:{jct_id}]` stanza, where `'{jct-id}'` refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

Options

number_of_bytes
The maximum number of unacknowledged bytes WebSEAL can accept per active multiplexed stream.

Usage

This stanza entry is optional.

Default value

None.

Example

```
http2-initial-window-size = 65535
```

http2-max-concurrent-streams

Use the **http2-max-concurrent-streams** stanza entry to set the maximum number of simultaneous multiplexed streams WebSEAL will accept per HTTP/2 network connection.

Syntax

```
http2-max-concurrent-streams = number_of_streams
```

Description

This stanza entry sets the maximum number of simultaneous multiplexed streams WebSEAL will accept per HTTP/2 network connection to a junctioned server.

Note:

- Each stream will have a **http2-initial-window-size** byte buffer.
- Each stream will need a worker-thread to process the one request or response that is sent over it before it is ended.

This configuration item may be customized for a particular junction by adding the adjusted configuration item to a `[junction:{jct_id}]` stanza, where `'{jct-id}'` refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

Options

number_of_steams
The maximum number of simultaneous multiplexed streams WebSEAL will accept per HTTP/2 network connection. A value of -1 denotes the unlimited setting and is not recommended in a production WebSEAL environment as memory use in WebSEAL would be unbounded.

Usage

This stanza entry is optional.

Default value

None.

Example

```
http2-max-concurrent-streams = 100
```

http2-max-frame-size

Use the **http2-max-frame-size** stanza entry to define the maximum size of the body of a single HTTP/2 protocol frame sent over the HTTP/2 network connection.

Syntax

```
http2-max-frame-size = number_of_bytes
```

Description

This stanza entry defines the maximum size of the body of a single HTTP/2 protocol frame sent over the HTTP/2 network connection to a junctioned server.

This configuration item may be customized for a particular junction by adding the adjusted configuration item to a `[junction:{jct_id}]` stanza, where `'{jct-id}'` refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

Options

number_of_bytes
The maximum size of the body of a single HTTP/2 protocol frame that can be sent over the HTTP/2 network connection.

Usage

This stanza entry is optional.

Default value

None.

Example

```
http2-max-frame-size = 16384
```

http2-max-header-list-size

Use the **http2-max-header-list-size** stanza entry to define the maximum size of headers that can be sent in a request on an HTTP/2 stream.

Syntax

```
http2-max-header-list-size = number_of_bytes
```

Description

This stanza entry defines the maximum size of headers in bytes that can be sent in a request on an HTTP/2 stream to a junctioned server. A value of -1 denotes the unlimited setting and is not recommended in a production WebSEAL environment as memory use in WebSEAL would be unbounded. If this entry is not set, it will default to the value of **[server] max-client-read**.

This configuration item may be customized for a particular junction by adding the adjusted configuration item to a `[junction:{jct_id}]` stanza, where *'{jct-id}'* refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

Options

number_of_bytes
The maximum size of headers that can be sent in a request on an HTTP/2 stream.

Usage

This stanza entry is optional.

Default value

The value of the **max-client-read** entry in the **[server]** stanza.

Example

```
http2-max-header-list-size = 32768
```

http-header-attributes

Use the `http-header-attributes` stanza entry to define the credential attributes which will be added as HTTP headers to the request.

Syntax

```
http-header-attributes = <attr-name>{:::<hdr-name>}
```

Description

Specifies which attributes from the credential will be inserted, as HTTP headers, into requests which are sent to the junctioned server.

This entry may be repeated multiple times, once for each attribute which is to be added.

This configuration item may be customized for a particular junction by adding the adjusted configuration item to a `[junction:{jct_id}]` stanza, where '`{jct-id}`' refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

Options

<attr-name>
The name of the credential attribute.

<hdr-name>
The name of the HTTP header which will contain the attribute. If the header name is not specified, the name of the attribute will be used as the HTTP header name.

Usage

This stanza entry is optional.

Default Value

None.

Example

```
http-header-attributes = AZN_CRED_PRINCIPAL_NAME:::principal
```

http-timeout

Syntax

```
http-timeout = number_of_seconds
```

Description

Integer value indicating the timeout, in seconds, for sending to and reading from a TCP junction.

You can customize this configuration item for a particular junction by adding the adjusted configuration item to a **[junction:*{junction_name}*]** stanza.

where *{junction_name}* refers to the junction point for a standard junction (including the leading / character) or the virtual host label for a virtual host junction.

Options

number_of_seconds
Integer value indicating the timeout, in seconds, for sending to and reading from a TCP junction. The minimum value is 0. When the value is 0, there is no timeout. WebSEAL does not impose a maximum value.

Usage

This stanza entry is required.

Default value

120

```
http-timeout = 120
```

https-timeout

Syntax

```
https-timeout = number_of_seconds
```

Description

Integer value indicating the timeout, in seconds, for sending to and reading from a Secure Socket Layer (SSL) junction.

You can customize this configuration item for a particular junction by adding the adjusted configuration item to a **[junction:*{junction_name}*]** stanza.

where *{junction_name}* refers to the junction point for a standard junction (including the leading / character) or the virtual host label for a virtual host junction.

Options

number_of_seconds
Integer value indicating the timeout, in seconds, for sending to and reading from a Secure Socket Layer (SSL) junction. The minimum value is 0. When the value is 0, there is no timeout. WebSEAL does not impose a maximum value.

Usage

This stanza entry is required.

Default value

120

```
https-timeout = 120
```

ignore-svc-unavailable

Use **ignore-svc-unavailable** to control whether WebSEAL handles a 503 'Service Unavailable' from a back-end server or returns it to the client.

Syntax

```
ignore-svc-unavailable = {true|false}
```

Description

The following configuration entry sets whether WebSEAL should handle a 503 "Service Unavailable" from a back-end server or return it to the client.

This configuration item might be customized for a particular junction by adding the adjusted configuration item to a **[junction:*{jct_id}*]** stanza, where *{jct-id}* refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

Type

Boolean

Usage

This stanza entry is optional.

Default Value

False

Example

```
ignore-svc-unavailable = false
```

insert-client-real-ip-for-option-r

Syntax

```
insert-client-real-ip-for-option-r = {yes|no}
```

Description

Determines whether to use the current IP address of the client or the one cached in the credentials at authentication time for the value passed in a header to junctions created with the -r option.

Options

yes
Use the current IP address of the client for the value passed in a header to junctions created with the -r option.

no
Use the client IP address cached in the credentials at authentication time for the value passed in a header to junctions created with the -r option.

Usage

This stanza entry is required.

Default value

no

Example

```
insert-client-real-ip-for-option-r = no
```

io-buffer-size

Syntax

```
io-buffer-size = number_of_bytes
```

Description

Positive integer value indicating the buffer size, in bytes, for low-level reads from and writes to a junction.

Options

number_of_bytes

Positive integer value indicating the buffer size, in bytes, for low-level reads from and writes to a junction.

The minimum value is 1. WebSEAL does not impose a maximum value.

A very small value (for instance, 10 bytes) can hurt performance by causing very frequent calls to the low-level read/write APIs. Up to a certain point, larger values improve performance because they correspondingly reduce the calls to the low-level I/O functions.

However, the low-level I/O functions may have their own internal buffers, such as the TCP send and receive buffers. Once **io-buffer-size** exceeds the size of those buffers (which are typically not large), there is no longer any performance improvement at all because those functions only read part of the buffer at the time.

Reasonable values for **io-buffer-size** range between 1 kB and 8 kB. Values smaller than this range causes calling the low-level I/O functions too frequently. Values larger than this range wastes memory. A 2 MB I/O buffer size uses 4 MB for each worker thread communicating with the junctioned server, since there is both an input and output buffer.

Usage

This stanza entry is required.

Default value

4096

Example

```
io-buffer-size = 4096
```

jct-cert-keyfile

Syntax

```
jct-cert-keyfile = file_name
```

Description

WebSEAL provides an option to configure a separate certificate key database for junction SSL operations rather than sharing the one used for client certificates specified in the **[ssl]** stanza. The **jct-cert-keyfile** parameter specifies the junction certificate keyfile. If this option is enabled, this is the keyfile used for CA and client certificates when negotiating SSL sessions with junctions.

Note: This stanza entry is commented out in the WebSEAL configuration file. To enable the option of using a separate certificate key database for junctioned servers, create the pd`jct.`kdb keyfile (and optional stash file) using iKeyman, and uncomment the options **jct-cert-keyfile** and **jct-cert-keyfile-stash** in the configuration file.

Options

file_name

The name of the optional, separate junction certificate keyfile.

Note: If **jct-cert-keyfile** is defined, **jct-cert-keyfile-stash** must also be defined.

Usage

This stanza entry is optional.

Default value

pdjct.kdb

Example

```
jct-cert-keyfile = pdjct.kdb
```

jct-cert-keyfile-stash

Syntax

```
jct-cert-keyfile-stash = file_name
```

Description

WebSEAL provides an option to configure a separate certificate key database for junction SSL operations rather than sharing the one used for client certificates specified in the **[ssl]** stanza. The **jct-cert-keyfile-stash** parameter specifies the stash file for the optional, separate junction certificate database.

Note: This stanza entry is commented out in the WebSEAL configuration file. To enable the option of using a separate certificate key database for junctioned servers, create the pdjct.kdb keyfile (and optional stash file) using iKeyman, and uncomment the options **jct-cert-keyfile** and **jct-cert-keyfile-stash** in the configuration file.

Options

file_name
The name of the stash file for the optional, separate junction certificate database.

Note: If **jct-cert-keyfile** is defined, **jct-cert-keyfile-stash** must also be defined.

Usage

This stanza entry is optional.

Default value

pdjct.sth

Example

```
jct-cert-keyfile-stash = pdjct.sth
```

jct-nist-compliance

Use the **jct-nist-compliance** stanza entry to enable or disable NIST SP800-131A compliance for junction connections.

Syntax

```
jct-nist-compliance = {yes|no}
```

Description

Enables or disables NIST SP800-131A compliance for junction connections.

Enabling NIST SP800-131A compliance results in the following automatic configuration:

- Enables FIPS mode processing.

  Note: When NIST SP800-131A compliance is enabled, FIPS mode processing is enabled regardless of the setting for the **fips-mode-processing** configuration entry.
- Enables TLS v1.2.

  Notes:

  – When NIST SP800-131A compliance is enabled, TLS v1.2 is enabled regardless of the setting for the **disable-tls-v12** configuration entry.

  – TLS v1 and TLS v1.1 are not disabled.
- Enables the appropriate signature algorithms.
- Sets the minimum RSA key size to 2048 bytes.

Options

yes
A value of yes enables NIST SP800-131A compliance.

no
A value of no disables NIST SP800-131A compliance.

Usage

This stanza entry is optional.

Default value

no

Example

```
jct-nist-compliance = no
```

jct-ocsp-enable

Syntax

```
jct-ocsp-enable = {yes|no}
```

Description

Enable Online Certificate Status Protocol (OCSP) for checking the revocation status of certificates supplied by a junction server using the OCSP URL embedded in the certificate using an Authority Information Access (AIA) extension.

Options

yes
Enable OCSP to check the revocation status of junction server supplied certificates.

no
Disable OCSP checking of junction server supplied certificates.

Usage

This stanza entry is optional.

Note: This option can be used as an alternative to, or in conjunction with, the **jct-ocsp-url** option.

Default value

no

Example

```
jct-ocsp-enable = no
```

jct-ocsp-max-response-size

Syntax

```
jct-ocsp-max-response-size = number of bytes
```

Description

Sets the maximum response size (in bytes) that will be accepted as a response from an OCSP responder. This limit helps protect against a denial of service attack.

Options

Maximum response size, in bytes.

Usage

This stanza entry is optional.

Default value

204080

Example

```
jct-ocsp-max-response-size = 20480
```

jct-ocsp-nonce-check-enable

Syntax

```
jct-ocsp-nonce-check-enable = {yes|no}
```

Description

Determines whether WebSEAL checks the nonce in the OCSP response. Enabling this option improves security but can cause OCSP Response validation to fail if there is a caching proxy between WebSEAL and the OCSP Responder. Note that enabling this option automatically enables the jct-ocsp-nonce-generation-enable option.

Options

yes
WebSEAL checks the nonce in the OCSP response to verify that it matches the nonce from the request.

no
WebSEAL does not check the nonce in the OCSP response.

Usage

This stanza entry is optional.

Default value

no

Example

```
jct-ocsp-nonce-check-enable = no
```

jct-ocsp-nonce-generation-enable

Syntax

```
jct-ocsp-nonce-generation-enable = {yes|no}
```

Description

Determines whether WebSEAL generates a nonce as part of the OCSP request. Enabling this option can improve security by preventing replay attacks on WebSEAL but may cause an excessive load on an OCSP Responder appliance as the responder cannot use cached responses and must sign each response.

Options

yes
WebSEAL generates a nonce as part of the OCSP request.

no
WebSEAL does not generate a nonce as part of the OCSP request.

Usage

This stanza entry is optional.

Default value

no

Example

```
jct-ocsp-nonce-generation-enable = no
```

jct-ocsp-proxy-server-name

Syntax

```
jct-ocsp-proxy-server-name = <proxy host name>
```

Description

Specifies the name of the proxy server that provides access to the OCSP responder.

Options

proxy host name
Fully qualified name of the proxy server.

Usage

This stanza entry is optional.

Default value

None

Example

```
jct-ocsp-proxy-server-name = proxy.ibm.com
```

jct-ocsp-proxy-server-port

Syntax

```
jct-ocsp-proxy-server-port = <proxy host port number>
```

Description

Specifies the port number of the proxy server that provides access to the OCSP Responder.

Options

proxy host port number
Port number used by the proxy server to route OCSP requests and responses.

Usage

This stanza entry is optional.

Default value

None

Example

```
jct-ocsp-proxy-server-port = 8888
```

jct-ocsp-url

Syntax

```
jct-ocsp-url = <OCSP Responder URL>
```

Description

Specifies the URL for the OCSP Responder. If a URL is provided, WebSEAL uses OCSP for all revocation status checking regardless of whether the certificate has an Authority Information Access (AIA) extension, which means that OCSP works with existing certificates. WebSEAL tries the OCSP Responder that is configured by this method first, rather than using a location specified by AIA extension. If revocation status is undetermined, and if **jct-ocsp-enable** is set to yes, then WebSEAL tries to obtain revocation status using the access method in the AIA extension.

Options

OCSP Responder URL
URL of the OCSP Responder.

Usage

This stanza entry is optional.

Default value

None

Example

```
jct-ocsp-url = http://responder.ibm.com/
```

jct-ssl-reneg-warning-rate

Syntax

```
jct-ssl-reneg-warning-rate = number_renegotiations/minute
```

Description

When this option is set to a value greater than zero (0), WebSEAL produces a warning message if the SSL session renegotiation rate between junction servers and WebSEAL reaches this level or greater. The value is specified as the number of renegotiations per minute.

Options

number_renegotiations/minute
Rate of session renegotiations between junction servers and WebSEAL.

Usage

This stanza entry is required.

Default value

0

Example

```
jct-ssl-reneg-warning-rate = 0
```

jct-undetermined-revocation-cert-action

Syntax

```
jct-undetermined-revocation-cert-action = {ignore | log | reject}
```

Description

Controls the action that WebSEAL takes if OCSP or CRL is enabled but the responder cannot determine the revocation status of a certificate (that is, the revocation status is unknown). The appropriate values for this entry should be provided by the OCSP or CRL Responder owner.

Options

ignore
WebSEAL ignores the undetermined revocation status and permits use of the certificate.

log
WebSEAL logs the fact that the certificate status is undetermined and permits use of the certificate.

reject
WebSEAL logs the fact that the certificate status is undetermined and rejects the certificate.

Usage

This stanza entry is optional.

Default value

log

Example

```
jct-undetermined-revocation-cert-action = log
```

jmt-map

Syntax

```
jmt-map = file_name
```

Description

The name of the file that contains the location of the Junction-to- Request Mapping Table (JMT).

The administrator can rename this file if necessary. The file name can be any file name valid for the operating system file system.

Options

file_name
Name of the file that contains the location of the Junction-to- Request Mapping Table (JMT).

Usage

This stanza entry is required.

Default value

`jmt.conf`

Example

```
jmt-map = jmt.conf
```

junction-specific-snoop

Use the `junction-specific-snoop` stanza entry to control whether junction specific trace points are enabled for 'snoop' tracing.

Syntax

```
junction-specific-snoop = {yes|true|no|false}
```

Description

When this option is enabled junction specific trace handles will be created for the pdweb.snoop.jct trace component. This allows an administrator to activate snoop tracing for a single junction. The trace name will be of the format: pdweb.snoop.jct.<jct-name>. When generating the trace name any '.' characters found in the junction name will be replaced with '_' and any non-ascii characters will be replaced with '?'.

Options

yes
Junction specific tracing is enabled.

true
Junction specific tracing is enabled.

no
Junction specific tracing is not enabled.

false
Junction specific tracing is not enabled.

Usage

This stanza entry is optional

Default Value

no

Example

```
junction-specific-snoop = yes
```

kerberos-keytab-file

Use the **kerberos-keytab-file** entry to set the name of the Kerberos key table file for the WebSEAL server.

Syntax

```
kerberos-keytab-file = keytab_file_name
```

Description

This stanza entry defines the Kerberos key table file for the WebSEAL server.

Options

keytab_file_name
The name of the Kerberos key table file.

Usage

This stanza entry is required when Kerberos SSO authentication for junctions is enabled.

Default value

None.

Example

```
kerberos-keytab-file = example.kdb
```

kerberos-principal-name

Use the **kerberos-principal-name** entry to set the service principal name of the impersonating user when creating a Kerberos token.

Syntax

```
kerberos-principal-name = principal_name
```

Description

The service principal name can be determined by executing the Microsoft utility **setspn**. For example:

```
setspn -L user
```

where *user* is the identity of the WebSEAL account.

This configuration item may be customized for a particular junction by adding the adjusted configuration item to a `[junction:{jct_id}]` stanza, where *'{jct-id}'* refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

Options

principal_name
The service principal name of the impersonating user when creating a Kerberos token.

Usage

This stanza entry is required when Kerberos SSO authentication for junctions is enabled.

Default value

None.

Example

```
kerberos-principal-name = HTTP/webseal@<realm>
```

kerberos-service-name

Use the **kerberos-service-name** entry to set the service principal name of the target.

Syntax

```
kerberos-service-name = service-name
```

Description

The service principal name can be determined by executing the Microsoft utility **setspn** (that is, `setspn -L user`, where `user` is the identity of the back-end Web server's account). This configuration item may be customized for a particular junction by adding the adjusted configuration item to a `[junction:{jct_id}]` stanza, where `'{jct-id}'` refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

The format is:

```
kerberos-service-name = HTTP/<username>@<realm>
```

Options

service-name
The service principal name of the target.

Usage

This stanza entry is required when Kerberos SSO authentication for junctions is enabled.

Default value

None.

Example

```
kerberos-service-name = HTTP/myservice@<realm>
```

kerberos-sso-enable

Use the **kerberos-sso-enable** entry to enable or disable SSO for junctions.

Syntax

```
kerberos-sso-enable = {yes|true|no|false}
```

Description

You can customize this configuration item for a particular junction by adding the adjusted configuration item to a **[junction:{junction_name}]** stanza.

Options

yes
Enable.

true
Enable.

no
Disable.

false
Disable.

Usage

This stanza entry is required.

Default value

no.

Example

```
kerberos-sso-enable = no
```

kerberos-user-identity

Use the **kerberos-user-identity** stanza entry to enable and define a custom user principal name (UPN). The custom UPN can be constructed from either plain text or the contents of credential attributes.

Syntax

```
kerberos-user-identity = username@domain
kerberos-user-identity = username
kerberos-user-identity = @domain
kerberos-user-identity = fqdn
```

Description

An administrator can overwrite the UPN or sections of the UPN for Kerberos constrained delegation users with this entry. The replacement information can be either plain text or names of credential attributes that store the required information. If you specify plain text, the text is directly copied into the UPN sections. If you specify names of credential attributes, the replacement text is fetched from the value of the corresponding credential attribute.

The domain information can also be extracted from the DC elements of the user's DN through the attribute **attr:dn**.

If no user name is defined, the client credential name is used.

If no domain is defined, the WebSEAL service account domain is used.

The domain value must be uppercase. Any input data that is not uppercase is automatically converted to uppercase. The domain must also be added as a realm to the Kerberos configuration.

Options

username@domain

Replaces both the user name and the domain separately.

username

Replaces only the user name. The WebSEAL service account domain is used as the user domain.

@domain

Replaces only the domain. The user name is obtained from the client credential.

fqdn

Replaces both the user name and domain with a single attribute. The value of this attribute must contain both the user name and the domain.

Usage

This stanza entry is optional. It can be customized for a particular junction in the **[junction:** *junction_name*] stanza.

Default value

None

Example

```
kerberos-user-identity = bob@IBM.COM
kerberos-user-identity = attr:SamAccountName@IBM.COM
kerberos-user-identity = @attr:dn
kerberos-user-identity = attr:FQDN
```

managed-cookies-list

Syntax

```
managed-cookies-list = list
```

Description

The managed-cookies-list contains a comma-separated list of patterns that will be matched against the names of cookies returned by junctioned servers. Cookies with names that match the patterns in this list are stored in the WebSEAL cookie jar and not returned to the client. Cookies that do not match these patterns are returned to the client browser.

The WebSEAL cookie jar is turned off by not specifying any cookies in the **managed-cookies-list**.

You can customize this configuration item for a particular junction by adding the adjusted configuration item to a **[junction:*{junction_name}*]** stanza.

where *{junction_name}* refers to the junction point for a standard junction (including the leading / character) or the virtual host label for a virtual host junction.

Options

list

A comma-separated list of pattern-matched cookie names.

Usage

This stanza entry is optional.

Default value

This option is empty by default.

```
managed-cookies-list = JSESS*,Ltpa*
```

mangle-domain-cookies

Syntax

```
mangle-domain-cookies = {yes | no}
```

Description

Enables or disables WebSEAL domain cookie name mangling behavior.

Note:

1. This option enables domain cookie mangling on a server-wide basis. The option cannot be configured on a per-junction basis.
2. This option is relevant only for junctions that use a reprocessing solution such as -j or JMT.
3. This option does not affect cookies listed in preserve-cookie-names.

Options

yes
Enables WebSEAL to mangle the names of domain cookies. Information identifying the junction is added to the cookie name, and the cookie is only associated with that junction. If **mangle-path-into-cookie-name** is set to yes, then the backend path attribute information is also mangled into the cookie name.

no
WebSEAL will not mangle the names of domain cookies.

Usage

This stanza entry is optional.

Default value

This option is disabled by default.

Example

```
mangle-domain-cookies = yes
```

match-vhj-first

Helps determine the order in which WebSEAL searches for a request in a standard or a virtual host junction table.

Syntax

```
match-vhj-first = {yes|no}
```

Description

WebSEAL manages separate junction tables for standard and virtual host junctions. When a request comes in, WebSEAL searches the virtual host junction table first. If WebSEAL does not find a match, it searches the table that manages standard junctions. The **match-vhj-first** configuration can reverse the search order so that WebSEAL searches the standard junction table before searching the virtual host junction table.

You can customize this configuration item for a particular junction by adding the adjusted configuration item to a **[junction:*{junction_name}*]** stanza, where *{junction_name}* refers to the junction point for a standard junction (including the leading / character) or the virtual host label for a virtual host junction.

Options

yes
WebSEAL searches the virtual host junction table first.

no
WebSEAL searches the standard junction table first.

Usage

This stanza entry is not optional.

Default value

yes

Example

The following example tells WebSEAL to search the standard junction table first:

```
match-vhj-first = no
```

max-cached-persistent-connections

Syntax

```
max-cached-persistent-connections = number_of_connections
```

Description

The maximum number of persistent connections that will be stored in the cache for future use. Connections with junctioned Web servers will be cached for future use unless the configured limit (as defined by this configuration entry) is reached, or unless the **connection:close** header is received in the HTTP response.

Note: If this setting is enabled, there is the potential for different user sessions to use the same connection when processing junction requests. To disable the persistent connection functionality, specify a **max-cached-persistent-connections** value of zero (`0`).

You can customize this configuration item for a particular junction by adding the adjusted configuration item to a **[junction:*{junction_name}*]** stanza.

where *{junction_name}* refers to the junction point for a standard junction (including the leading / character) or the virtual host label for a virtual host junction.

Options

number_of_connections
Integer value indicating the maximum number of persistent connections that will be stored in the cache for future use. A value of zero (`0`) disables this support. WebSEAL imposes no maximum on this value.

Usage

This stanza entry is required.

Default value

`0`

```
max-cached-persistent-connections = 0
```

max-jct-read

Use the **`max-jct-read`** stanza entry to control the amount of header data WebSEAL will read from responses.

Syntax

```
max-jct-read = number_of_bytes
```

Description

Maximum size, in bytes, of headers WebSEAL read from responses. By default, WebSEAL read headers up to 65536 bytes in length. When larger headers are expected, for example in the case of an EAI authentication where the user belongs to many groups, this parameter must be increased in order for WebSEAL to parse the complete header.

You can customize this configuration item for a particular junction by adding the adjusted configuration item to a **`[junction:{junction_name}]`** stanza, where *{junction_name}* refers to the junction point for a standard junction (including the leading /character) or the virtual host label for a virtual host junction.

Options

number_of_bytes
The maximum number of bytes of header data WebSEAL will read from responses.

Usage

This stanza entry is optional.

Default value

65536

Example

```
max-jct-read = 131072
```

max-webseal-header-size

Syntax

```
max-webseal-header-size = number_of_bytes
```

Description

Integer value indicating the maximum size, in bytes, of HTTP headers generated by the WebSEAL server. Headers greater in size that this value are split across multiple HTTP Headers.

Note: The **max-webseal-header-size** entry does not limit the maximum size of HTTP-Tag-Value headers.

Options

number_of_bytes
Integer value indicating the maximum size, in bytes, of HTTP headers generated by the WebSEAL server. A value of zero (0) disables this support. WebSEAL imposes no maximum on this value.

Usage

This stanza entry is required.

Default value

0

Example

```
max-webseal-header-size = 0
```

pass-http-only-cookie-attr

Syntax

```
pass-http-only-cookie-attr = {yes|no}
```

Description

Indicates whether WebSEAL will pass or remove the HTTP-only attribute from the Set-Cookie headers sent by junctioned servers.

Options

yes
Enables WebSEAL to pass the HTTP-only attribute from Set-Cookie headers sent by junctioned servers.

no
Enables WebSEAL to remove the HTTP-only attribute from Set-Cookie headers sent by junctioned servers.

Usage

This stanza entry is required.

Default value

yes

Example

```
pass-http-only-cookie-attr = yes
```

persistent-con-timeout

Syntax

```
persistent-con-timeout = number_of_seconds
```

Description

Indicates the maximum number of seconds a persistent connection can remain idle in a cache before the connection is cleaned up and closed by WebSEAL.

Use an integer value lower than the configured maximum connection lifetime for the junctioned web server. For example, the connection lifetime for a junctioned Apache web server is controlled by the **KeepAliveTimeout** configuration entry.

You can customize the **persistent-con-timeout** configuration item for a particular junction by adding the adjusted configuration item to a **[junction:*{junction_id}*]** stanza.

where *{junction_id}* refers to the junction point for a standard junction (including the leading / character) or the virtual host label for a virtual host junction.

Note: If you do not use an integer value lower than the connection lifetime on the junctioned web server, you might encounter the following problem.

If the **[junction] max-cached-persistent-connections** configuration entry is set to a value greater than zero, WebSEAL reuses its TCP/IP session with the junctioned back-end server. If the junctioned back-end server closes the socket at the same time that WebSEAL starts to use this session to send a request, the request fails.

To send the request again, WebSEAL opens a new TCP/IP session. If the request body is larger than the size that WebSEAL can cache, WebSEAL fails to resend the request and generates a 500 error.

Options

number_of_seconds
: Integer value that indicates the maximum number of seconds a persistent connection can remain idle in a cache before the connection is closed by WebSEAL. The minimum value is 1. WebSEAL does not impose a maximum value.

Usage

This stanza entry is required.

Default value

5

Example

```
persistent-con-timeout = 5
```

ping-method

Syntax

```
ping-method = method
```

Description

The WebSEAL server performs a periodic background ping of each junctioned Web server, to determine whether it is running. The optional **ping-method** entry sets the HTTP request type used in these pings. The valid options include any valid HTTP request method (for example, *HEAD* or *GET*, for HTTP HEAD and HTTP GET requests respectively).

This configuration item may be customized for a particular junction by adding the adjusted configuration item to a **[junction:*{junction_name}*]** stanza.

where *{junction_name}* refers to the junction point for a standard junction (including the leading / character) or the virtual host label for a virtual host junction.

Options

method
Perform a HTTP request using the specified method to determine the state of the junctioned server.

Usage

This stanza entry is optional.

Default value

HEAD

Example

```
ping-method = GET
```

ping-response-code-rules

Use the **ping-response-code-rules** configuration entry to define the rules that are used to determine whether the HTTP status code of the ping responses indicate a healthy or an unhealthy junctioned Web server.

Syntax

```
ping-response-code-rules = list
```

Description

The WebSEAL server performs a periodic background ping of each junctioned Web server, to determine whether the junctioned Web server is running. The optional **ping-response-code-rules** configuration entry defines the rules that are used to determine whether the HTTP status code of the ping responses indicate a healthy or an unhealthy junctioned Web server.

If valid values are configured for both **ping-response-code-rules** and **response-code-rules**, the specified **ping-response-code-rules** will be applied to the ping requests initiated by WebSEAL, and other requests will be matched against **response-code-rules** to determine the server state.

If a valid **ping-response-code-rules** value is configured but **response-code-rules** is not, the specified **ping-response-code-rules** will be applied to the ping requests initiated by WebSEAL, and other requests will not be used to determine the server state. In this case, **ping-response-code-rules** are the only rules used to determine the server state.

If the **ping-response-code-rules** configuration entry is not set, the rules that are specified by the **response-code-rules** configuration entry will also apply to ping requests.

If the **ping-attempt-threshold** entry is configured, when the junction is marked as running, it must fail this number of consecutive ping requests before it is marked as not running. If this entry is not set it is default to 1.

If the **recovery-ping-attempt-threshold** entry is configured, when the junction is marked as not running, it must return this number of consecutive successful recovery ping responses before it is marked as running. If this entry is not set it is default to 1.

The configuration entry contains a space separated list of rules. Each rule has the format: `[+|-]<code>` (e.g. `-50?`)

where:

+
: Indicates that this is a healthy response code.

-
: Indicates that this is an unhealthy response code.

<code>
: The corresponding response code, which can also contain pattern matching characters such as `*` and `?`

The HTTP response codes are evaluated against each rule in sequence until a match is found. The corresponding code (+ | -) determines whether the junctioned Web server is healthy or not.If the response code matches no configured rules, the junctioned Web server is considered healthy.

You can customize this configuration item for a particular junction by adding the adjusted configuration item to a **[junction:*{junction_name}*]** stanza, where *{junction_name}* refers to the junction point for a standard junction (including the leading / character) or the virtual host label for a virtual host junction.

Options

list
: A space separated list of response code rules. These rules determine whether the ping response from a junctioned Web server indicates a healthy or an unhealthy server.

Usage

This stanza entry is optional.

Default value

None.

Example

```
ping-response-code-rules = +2?? -*
```

ping-attempt-threshold

Use this entry to define the number of consecutive failed ping requests before the junctioned server will be marked as not running.

Syntax

```
ping-attempt-threshold = number
```

Description

The WebSEAL server performs a periodic background ping of each junctioned Web server, to determine whether it is running.

If this entry is configured, when the junction is marked as running, it must fail this number of consecutive ping requests before it will be marked as not running.

If this entry is not set it will default to 1.

You can customize this configuration item for a particular junction by adding the adjusted configuration item to a **[junction:*{junction_name}*]** stanza, where *{junction_name}* refers to the junction point for a standard junction (including the leading / character) or the virtual host label for a virtual host junction.

Options

number
The number of consecutive failed ping requests before it will be marked as not running.

Usage

This stanza entry is optional.

Default value

1

Example

```
ping-attempt-threshold = 1
```

ping-time

Syntax

```
ping-time = number_of_seconds
```

Description

Integer value indicating the number of seconds between pings issued by the WebSEAL server. The pings are issued periodically in the background to verify that junctioned WebSEAL servers are running.

If the server is deemed not running, the **recovery-ping-time** value determines the interval at which pings are sent until the server is running. The type of ping used is determined by the **ping-method** value. HTTP response code rules can be defined using the **response-code-rules** configuration entry.

You can customize this configuration item for a particular junction by adding the adjusted configuration item to a **[junction:*{junction_name}*]** stanza, where *{junction_name}* refers to the junction point for a standard junction (including the leading / character) or the virtual host label for a virtual host junction.

Options

number_of_seconds
Integer value indicating the number of seconds between pings issued by the WebSEAL server. The minimum value is 1. WebSEAL does not impose a maximum value.

Usage

To turn this ping off, set this entry to zero. If this entry is set to zero, the **recovery-ping-time** must be set.

Default value

300

Example

```
ping-time = 300
```

ping-timeout

Use this entry to set a different timeout value for the 'ping' operations.

Syntax

```
ping-timeout = timeout
```

Description

Timeout (in seconds) for sending ping requests to, and reading ping responses from, the junction. The value must be an integer greater than or equal to zero. A value of zero causes WebSEAL to wait indefinitely. If no value is set the standard junction timeout values apply to the ping requests. This configuration item might be customized for a particular junction by adding the adjusted configuration item to a `[junction:{jct_id}]` stanza, where `'{jct-id}'` refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

If this entry is not set, it defaults to the value of the `http-timeout` or `https-timeout` configuration entries, depending on the type of junction that is being accessed.

Options

timeout
Timeout (in seconds) for sending ping requests to, and reading ping responses from, the junction

Usage
This stanza entry is optional.

Default value
The value defaults to the value of the `http-timeout` or `https-timeout` configuration entries, depending on the type of junction that is being accessed.

Example

```
ping-timeout = 30
```

ping-uri

Syntax

```
ping-uri = uri
```

Description

The WebSEAL server performs a periodic background ping of each junctioned Web server to determine whether it is running. The optional **ping-uri** configuration entry defines the URI that is accessed by the ping request. The defined URI is relative to the root Web space of the junctioned Web server. If the URI is missing, this value defaults to a /.

You can customize this configuration item for a particular junction by adding the adjusted configuration item to a **[junction:*{junction_name}*]** stanza.

where *{junction_name}* refers to the junction point for a standard junction (including the leading / character) or the virtual host label for a virtual host junction.

Options

uri
The URI that is accessed by the ping request.

Usage

This stanza entry is optional.

Default value

/

```
ping-uri = /apps/status
```

recovery-ping-time

Syntax

```
recovery-ping-time = number_of_seconds
```

Description

The WebSEAL server performs a periodic background ping of each junctioned Web server, to determine whether it is running. This entry sets the interval, in seconds, between pings when the server is determined to be not running.

You can customize this configuration item for a particular junction by adding the adjusted configuration item to a **[junction:*{junction_name}*]** stanza, where *{junction_name}* refers to the junction point for a standard junction (including the leading / character) or the virtual host label for a virtual host junction.

Options

number_of_seconds
Integer value indicating the number of seconds between pings issued by the WebSEAL server to a junctioned server that is determined to be not running. The minimum value is 1. WebSEAL does not impose a maximum value.

Usage

If this entry is not set, the **recovery-ping-time** defaults to the **ping-time** value.

Default value

300

Example

```
recovery-ping-time = 300
```

recovery-ping-attempt-threshold

Use this entry to define the number of consecutive successful recovery ping responses before a stopped junctioned server will be marked as running.

Syntax

```
recovery-ping-attempt-threshold = number
```

Description

The WebSEAL server performs a periodic background ping of each junctioned Web server, to determine whether it is running.

If this entry is configured, when the junction is marked as not running, it must return this number of consecutive successful recovery ping responses before it will be marked as running.

If this entry is not set it will default to 1.

You can customize this configuration item for a particular junction by adding the adjusted configuration item to a **[junction:*{junction_name}*]** stanza, where *{junction_name}* refers to the junction point for a standard junction (including the leading / character) or the virtual host label for a virtual host junction.

Options

number
The number of consecutive successful recovery ping responses before it will be marked as running.

Usage

This stanza entry is optional.

Default value

1

Example

```
recovery-ping-attempt-threshold = 1
```

reprocess-root-jct-404s

Syntax

```
reprocess-root-jct-404s = {yes|no}
```

Description

Used to reprocess requests for root junction resources that result in an HTTP 404 error.

The **dont-reprocess-jct-404s** entry (also in the **[junction]** stanza) can be set to yes to avoid multiple attempts to prepend a junction point to the beginning of the URL string when reprocessing requests that have resulted in an HTTP 404 status code.

WebSEAL determines whether the request is already known to be for a non-local junction.However, WebSEAL fails to add a junction point when requests have been made for a root junction created at "/". To modify this behavior and cause requests for root junction resources that result in an HTTP 404 error to be reprocessed, you can use this **reprocess-root-jct-404s** stanza entry.

Options

yes
Cause requests for root junction resources that result in an HTTP 404 error to be reprocessed regardless of the setting of the **dont-reprocess-jct-404s** entry (also in the **[junction]** stanza).

no
The value for the **dont-reprocess-jct-404s** entry (also in the **[junction]** stanza) will determine whether root junction requests that result in an HTTP 404 error are reprocessed. That is, if the value for **dont-reprocess-jct-404s** is no then the HTTP 404 errors will still be reprocessed.

Usage

This stanza entry is optional.

Default value

no

Example

```
reprocess-root-jct-404s = yes
```

reset-cookies-list

Syntax

```
reset-cookies-list = list
```

Description

Determines which cookies are reset when the user session is logged out. The request received from the client and the response sent back to the client are both examined for matching cookies.

You can customize this configuration item for a particular junction by adding the adjusted configuration item to a **[junction:*{junction_name}*]** stanza.

where *{junction_name}* refers to the junction point for a standard junction (including the leading / character) or the virtual host label for a virtual host junction.

Options

list
A comma-separated list of patterns. WebSEAL will reset any cookies with names that match the patterns in this list.

Usage

This stanza entry is required.

Default value

None.

```
reset-cookies-list = JSESS*,Ltpa*
```

response-code-rules

When a response of a client-initiated request is returned from the junctioned server, the optional **response-code-rules** configuration entry defines the rules that are used to determine from the HTTP status code of the responses whether the junctioned Web server is in a healthy or an unhealthy state.

Syntax

```
response-code-rules = list
```

Description

The optional **response-code-rules** configuration entry defines the rules that are used to determine whether HTTP responses indicate a healthy or an unhealthy junctioned Web server.

This configuration entry will apply to all requests if the **ping-response-code-rules** configuration entry has not been set. Otherwise, it will only apply to all client-initiated requests.

If this configuration entry is empty, and a **ping-response-code-rules** configuration has been specified, the response code received from client-initiated requests will not impact the health of the junction.

The configuration entry contains a space separated list of rules. Each rule has the format: `[+|-]<code>` (e.g. `-50?`)

where:

+
: Indicates that this is a healthy response code.

-
: Indicates that this is an unhealthy response code.

<code>
: The corresponding response code, which can also contain pattern matching characters such as `*` and `?`

The HTTP response codes are evaluated against each rule in sequence until a match is found. The corresponding code (+ | -) determines whether the junctioned Web server is healthy or not. If the response code matches no configured rules, the junctioned Web server is considered healthy.

You can customize this configuration item for a particular junction by adding the adjusted configuration item to a **[junction:*{junction_name}*]** stanza, where *{junction_name}* refers to the junction point for a standard junction (including the leading / character) or the virtual host label for a virtual host junction.

Options

list
: A space separated list of response code rules. These rules determine whether the response from a junctioned Web server indicates a healthy or an unhealthy server.

Usage

This stanza entry is optional.

Default value

None.

Example

```
response-code-rules = +2?? -*
```

share-cookies

Syntax

```
share-cookies = {yes|no}
```

Description

The **share-cookies** item is used to control whether the cookie jar will be shared across different junctions or whether each junction will have a dedicated cookie jar.

Options

yes
If this entry is set to *yes*, cookies will be sent over all junctions, regardless of the junction from which the cookie originated.

no
If this entry is set to *no*, only cookies received from the junction will be sent in requests to that junction.

Usage

This stanza entry is required.

Default value

no

Example

```
share-cookies = yes
```

server-hostname-validation

Use the `server-hostname-validation` stanza entry to control whether WebSEAL performs hostname validation on server certificates presented by Junctioned servers.

Syntax

```
server-hostname-validation = {disabled|critical|warning}
```

Description

Specifies whether hostname validation will be performed on the server certificates which are presented by junctioned servers. If enabled, the DNS hostname of the configured server will be checked against the CN and SAN fields of the server certificate.

If the expected CN is specified, using the '-0' option, during junction creation the hostname validation will not be performed.

You can customize this configuration item for a particular junction by adding the adjusted configuration item to a `[junction:{junction_name}]` stanza, where *{junction_name}* refers to the junction point for a standard junction (including the leading / character) or the virtual host label for a virtual host junction.

Options

`disabled`
No hostname validation will take place. This is the default.

`critical`
Hostname validation will take place and connections will be rejected if the validation fails.

`warning`
Hostname validation will take place, but connections will still be allowed if the validation fails. A warning message will however be displayed.

Usage

This stanza entry is optional.

Default value

`default`

Example

```
server-hostname-validation = critical
```

support-virtual-host-domain-cookies

Syntax

```
support-virtual-host-domain-cookies = {yes|no}
```

Description

If **allow-backend-domain-cookies** is set to `yes`, then this option modifies how WebSEAL validates the domain. This option has no effect if `validate-backend-domain-cookies = no`.

You can customize this configuration item for a particular junction by adding the adjusted configuration item to a **[junction:*{junction_name}*]** stanza.

where *{junction_name}* refers to the junction point for a standard junction (including the leading / character) or the virtual host label for a virtual host junction.

Options

yes
If set to "yes" then the domain cookie is validated by comparing it with the virtual host specified for a backend server with the **-v** junction option.

no
If set to "no", or if no virtual host was specified for a junction, then the fully qualified host name is compared with the domain value of a backend cookie for validation.

Usage

This stanza entry is required.

Default value

yes

```
support-virtual-host-domain-cookies = yes
```

use-new-stateful-on-error

Syntax

```
use-new-stateful-on-error = {yes|no}
```

Description

Control how WebSEAL responds to a stateful server that becomes unavailable.

This configuration item may be customized for a particular junction by adding the adjusted configuration item to a `[junction:{junction_name}]` stanza.

where *{junction_name}* refers to the junction point for a standard junction (including the leading / character) or the virtual host label for a virtual host junction. For example:

```
[junction:/WebApp]
```

Options

yes
When set to "yes" and the original server becomes unavailable during a session, WebSEAL directs the user's next request (containing the original stateful cookie) to a new replica server on the same stateful junction. If a new replica server is found on that stateful junction, and is responsive to the request, WebSEAL sets a new stateful cookie on the user's browser. Subsequent requests during this same session (and containing the new stateful cookie) are directed to this same new server.

no
When set to "no" and the original server becomes unavailable during a session, WebSEAL does not direct the user's subsequent requests to a new replica server on the same stateful junction. Instead, WebSEAL returns an error and attempts to access the same server for subsequent requests by the user during this session.

Usage

This stanza entry is required.

Default value

no

Example

```
use-new-stateful-on-error = yes
```

use-legacy-cookiejar-behavior

Use this configuration entry to allow legacy cookie jar behavior.

Syntax

```
use-legacy-cookiejar-behavior = {true|false}
```

Description

`use-legacy-cookiejar-behavior` must be set to `false` if the `use use-legacy-cookiejar-behavior-pdstateful` is used. See use-legacy-cookiejar-behavior-pdstateful.

Options

True
When set to `True`, WebSEAL generates PD_STATEFUL cookies after updating the cookie jar.

False
When set to `False`, WebSEAL does not generate PD_STATEFUL cookies after updating the cookie jar.

Usage

This stanza entry is required.

Default Value

`True`

Example

```
use-legacy-cookiejar-behavior = true
```

use-legacy-cookiejar-behavior-pdstateful

Use this configuration entry to allow legacy cookie jar behavior.

Syntax

```
use-legacy-cookiejar-behavior-pdstateful = {true|false}
```

Description

If `use-legacy-cookiejar-behavior-pdstateful` is set to `true`, WebSEAL generates PD_STATEFUL cookies after updating the cookie jar.

If `use-legacy-cookiejar-behavior-pdstateful = yes`, `use-legacy-cookiejar-behavior` must be set to `false`. See use-legacy-cookiejar-behavior.

Options

True
When set to `True`, WebSEAL generates PD_STATEFUL cookies after updating the cookie jar.

False
When set to `False`, WebSEAL does not generate PD_STATEFUL cookies after updating the cookie jar.

Usage

This stanza entry is required.

Default Value

`False`

Example

```
use-legacy-cookiejar-behavior-pdstateful = true
```

validate-backend-domain-cookies

Syntax

```
validate-backend-domain-cookies = {yes|no}
```

Description

Specifies how WebSEAL validates the domain.

You can customize this configuration item for a particular junction by adding the adjusted configuration item to a **[junction:*{junction_name}*]** stanza.

where *{junction_name}* refers to the junction point for a standard junction (including the leading / character) or the virtual host label for a virtual host junction.

Options

yes
If set to "yes" then domain cookies that adhere to the cookie specification are forwarded to the user. If the fully qualified host name of the originating back-end machine is the domain, then the cookie is forwarded to the user with no domain specified.

no
If set to "no", then all domain cookies are forwarded to the user, regardless of their content.

Usage

This stanza entry is required.

Default value

`yes`

```
validate-backend-domain-cookies = yes
```

worker-thread-hard-limit

Syntax

```
worker-thread-hard-limit = number_of_threads
```

Description

Integer value indicating the limit, expressed as a percentage, of the total worker threads that are to be used for processing requests for junctions.

Options

number_of_threads

Integer value indicating the limit, expressed as a percentage, of the total worker threads that are to be used for processing requests for junctions. The default value of 100 means that there is no limit.

When the value of **worker-thread-hard-limit** is less than 100, and the limit is exceeded, WebSEAL generates an error message.

Usage

This stanza entry is required.

Default value

100

Example

```
worker-thread-hard-limit = 100
```

worker-thread-soft-limit

Syntax

```
worker-thread-soft-limit = number_of_threads
```

Description

Integer value indicating the limit, expressed as a percentage, of the total worker threads that are to be used for processing requests for junctions.

Options

number_of_threads

Integer value indicating the limit, expressed as a percentage, of the total worker threads that are to be used for processing requests for junctions.

When the value of **worker-thread-soft-limit** is less than 100, and the limit is exceeded, WebSEAL generates a warning message.

Usage

This stanza entry is required.

Default value

90

Example

```
worker-thread-soft-limit = 90
```

[junction:<jct-id>] stanza

allow-backend-domain-cookies

Use the **allow-backend-domain-cookies** stanza entry to control whether WebSEAL sends domain cookies from a back-end server to a client.

Syntax

```
allow-backend-domain-cookies = {yes|no}
```

Description

Indicates whether WebSEAL can send domain cookies from a back-end server to a client.

You can customize this configuration item for a particular junction by adding the adjusted configuration item to a **[junction:*{junction_name}*]** stanza.

where *{junction_name}* refers to the junction point for a standard junction (including the leading / character) or the virtual host label for a virtual host junction.

Options

yes
WebSEAL is able to send domain cookies from a back-end server to a client.

no
WebSEAL is not able to send domain cookies from a back-end server to a client.

Usage

This stanza entry is required.

Default value

no

Example

```
allow-backend-domain-cookies = no
```

always-send-kerberos-tokens

Indicates whether WebSEAL sends a security token for every HTTP request or whether WebSEAL waits for a 401 response before it adds the security token.

Syntax

```
always-send-kerberos-tokens = {yes|true|no|false}
```

Description

You can customize this configuration item for a particular junction by adding the adjusted configuration item to a **[junction:{junction_name}]** stanza.

Options

yes
Enable.

true
Enable.

no
Disable.

false
Disable.

Usage

This stanza entry is required if Kerberos SSO authentication for junctions is enabled.

Default value

no.

Example

```
always-send-kerberos-tokens = no
```

connect-timeout

Syntax

```
connect-timeout = number_of_seconds
```

Description

Integer value indicating the timeout, in seconds, for establishing a connection to a junctioned server.

You can customize this configuration item for a particular junction by adding the adjusted configuration item to a **[junction:*{junction_name}*]** stanza.

where *{junction_name}* refers to the junction point for a standard junction (including the leading / character) or the virtual host label for a virtual host junction.

Options

number_of_seconds
Integer value indicating the timeout, in seconds, for establishing a connection to a junctioned server. The minimum value is 0. When the value is 0, there is no timeout. WebSEAL does not impose a maximum value.

Usage

This stanza entry is required.

Default value

30

```
connect-timeout = 30
```

disable-tls-v1

Use the **disable-tls-v1** entry in the **[junction]** stanza to control whether WebSEAL supports Transport Layer Security (TLS) version 1 for junction connections.

Syntax

```
disable-tls-v1 = {yes|no}
```

Description

Disables support for TLS Version 1 for junction connections. Support for TLS V1 is enabled by default.

This configuration item may be customized for a particular junction by adding the adjusted configuration item to a `[junction:{junction_name}]` stanza, where *{junction_name}* refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

Options

yes
The value yes means support is disabled.

no
The value no means the support is enabled.

Usage

This stanza entry is optional. When not specified, the default is no. The WebSEAL configuration sets this value.

Default value

no

Example

```
disable-tls-v1 = no
```

disable-ssl-v2

Use the **disable-ssl-v2** entry in the **[junction]** stanza to control whether WebSEAL supports SSL version 2 for junction connections.

Syntax

```
disable-ssl-v2 = {yes|no}
```

Description

Disables support for SSL Version 2 for junction connections. Support for SSL v2 is disabled by default.

This configuration item may be customized for a particular junction by adding the adjusted configuration item to a `[junction:{junction_name}]` stanza, where *{junction_name}* refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

Options

yes
The value `yes` means support is disabled.

no
The value no means the support is enabled.

Usage

This stanza entry is optional. When not specified, the default is `yes`. The WebSEAL configuration sets this value.

Default value

`yes`

Example

```
disable-ssl-v2 = yes
```

disable-ssl-v3

Use the **`disable-ssl-v3`** entry in the **`[junction]`** stanza to control whether WebSEAL supports SSL version 3 for junction connections.

Syntax

```
disable-ssl-v3 = {yes|no}
```

Description

Disables support for SSL Version 3 for junction connections. Support for SSL V3 is enabled by default.

This configuration item may be customized for a particular junction by adding the adjusted configuration item to a `[junction:{junction_name}]` stanza, where *{junction_name}* refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

Options

yes
The value `yes` means support is disabled.

no
The value no means the support is enabled

Usage

This stanza entry is optional. When not specified, the default is no. The WebSEAL configuration sets this value.

Default value

no

Example

```
disable-ssl-v3 = no
```

disable-tls-v11

Use the **disable-tls-v11** entry in the **[junction]** stanza to control whether WebSEAL supports Transport Layer Security (TLS) version 1.1 for junction connections.

Syntax

```
disable-tls-v11 = {yes|no}
```

Description

Determines whether WebSEAL supports TLS version 1.1 for junction connections. Support for TLS v1.1 is enabled by default.

This configuration item may be customized for a particular junction by adding the adjusted configuration item to a `[junction:{junction_name}]` stanza, where *{junction_name}* refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

Options

yes
The value yes disables support for TLS version 1.1.

no
The value no enables support for TLS version 1.1.

Usage

This stanza entry is optional. If this entry is not specified, the default is no.

Default value

no

Example

```
disable-tls-v11 = no
```

disable-tls-v12

Use the **disable-tls-v12** entry in the **[junction]** stanza to control whether WebSEAL supports Transport Layer Security (TLS) version 1.2 for junction connections.

Syntax

```
disable-tls-v12 = {yes|no}
```

Description

Determines whether WebSEAL supports Transport Layer Security (TLS) version 1.2 for junction connections. Support for TLS v1.2 is enabled by default.

This configuration item may be customized for a particular junction by adding the adjusted configuration item to a `[junction:{junction_name}]` stanza, where *{junction_name}* refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

Options

yes
The value `yes` disables support for TLS version 1.2.

no
The value `no` enables support for TLS version 1.2.

Usage

This stanza entry is optional. If this entry is not specified, the default is `no`.

Default value

`no`

Example

```
disable-tls-v12 = no
```

disable-tls-v13

Use the `disable-tls-v13` entry in the `[junction]` stanza to control whether support for TLS version 1.3 is enabled in WebSEAL.

Syntax

```
disable-tls-v13 = {yes|no}
```

Description

Determines whether WebSEAL supports Transport Layer Security (TLS) version 1.3 for junction connections. Support for TLS version 1.3 is disabled by default.

This configuration item may be customized for a particular junction by adding the adjusted configuration item to a `[junction:{junction_name}]` stanza, where *{junction_name}* refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

Options

yes
Disables support for TLS version 1.3

no
Enables support for TLS version 1.3

Usage

This stanza entry is optional. If this entry is not specified, the default is `yes`.

disable-tls-v13=no disables DPWNS0301W messages.

Default value

`yes`

Example

```
disable-tls-v13 = no
```

dynamic-addresses

Use the **dynamic-addresses** stanza entry to control whether the junction server host name is resolved to its IP address immediately before every communication with the junction server.

Syntax

```
dynamic-addresses = {yes|no}
```

Description

Indicates when the junction server host name is resolved to its corresponding IP address and used in communication with the junction server.

You can customize this configuration item for a particular junction by adding the adjusted configuration item to a **[junction:*{junction_name}*]** stanza.

where *{junction_name}* refers to the junction point for a standard junction (including the leading / character) or the virtual host label for a virtual host junction.

Options

yes
The junction server host name is resolved to its corresponding IP address immediately before any communication with the junction server.

If this configuration entry is set to yes, you can use the **dynamic-addresses-ttl** configuration entry to specify the length of time (in seconds) that a resolved IP address will be cached before it is discarded and another name resolution is attempted (time-to-live).

no
The junction server host name is resolved to its corresponding IP address and this address is used for subsequent communication with the junction server.

Usage

This stanza entry is required.

Default value

no

Example

```
dynamic-addresses = no
```

dynamic-addresses-ttl

Use the **dynamic-addresses-ttl** stanza entry to specify the length of time (in seconds) that a resolved IP address will be cached before it is discarded and another name resolution is attempted (time-to-live).

Syntax

```
dynamic-addresses-ttl = seconds
```

Description

If the **dynamic-addresses** configuration entry is set to yes, this configuration entry specifies the length of time (in seconds) that a resolved IP address will be cached before it is discarded and another name resolution is attempted (time-to-live).

You can customize this configuration item for a particular junction by adding the adjusted configuration item to a **[junction:*{junction_name}*]** stanza, where *{junction_name}* refers to the junction point for a standard junction (including the leading / character) or the virtual host label for a virtual host junction.

Options

seconds
The length of time (in seconds) that a resolved IP address will be cached before it is discarded and another name resolution is attempted (time-to-live).

Usage

This stanza entry is optional.

Default value

None

Example

```
dynamic-addresses-ttl = 0
```

http2-header-table-size

Use the **http2-header-table-size** stanza entry to define the max header table size for an HTTP/2 network connection.

Syntax

```
http2-header-table-size = table_size
```

Description

This stanza entry defines the maximum size in bytes that WebSEAL accepts for header compression table (RFC 7541). There is one table per HTTP/2 network connection to a junctioned server.

This configuration item may be customized for a particular junction by adding the adjusted configuration item to a [junction:*{jct_id}*] stanza, where *'{jct-id}'* refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

Options

table_size
The maximum size in bytes that WebSEAL will accept for header compression table.

Usage

This stanza entry is optional.

Default value

None.

Example

```
http2-header-table-size = 4096
```

http2-initial-window-size

Use the **http2-initial-window-size** stanza entry to define the maximum number of unacknowledged bytes WebSEAL can accept per active multiplexed stream.

Syntax

```
http2-initial-window-size = number_of_bytes
```

Description

This stanza entry defines the maximum number of unacknowledged bytes WebSEAL can accept per active multiplexed stream to a junctioned server.

This configuration item may be customized for a particular junction by adding the adjusted configuration item to a `[junction:{jct_id}]` stanza, where `'{jct-id}'` refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

Options

number_of_bytes
The maximum number of unacknowledged bytes WebSEAL can accept per active multiplexed stream.

Usage

This stanza entry is optional.

Default value

None.

Example

```
http2-initial-window-size = 65535
```

http2-max-concurrent-streams

Use the **http2-max-concurrent-streams** stanza entry to set the maximum number of simultaneous multiplexed streams WebSEAL will accept per HTTP/2 network connection.

Syntax

```
http2-max-concurrent-streams = number_of_streams
```

Description

This stanza entry sets the maximum number of simultaneous multiplexed streams WebSEAL will accept per HTTP/2 network connection to a junctioned server.

Note:

- Each stream will have a **http2-initial-window-size** byte buffer.
- Each stream will need a worker-thread to process the one request or response that is sent over it before it is ended.

This configuration item may be customized for a particular junction by adding the adjusted configuration item to a `[junction:{jct_id}]` stanza, where `'{jct-id}'` refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

Options

number_of_steams
The maximum number of simultaneous multiplexed streams WebSEAL will accept per HTTP/2 network connection. A value of -1 denotes the unlimited setting and is not recommended in a production WebSEAL environment as memory use in WebSEAL would be unbounded.

Usage

This stanza entry is optional.

Default value

None.

Example

```
http2-max-concurrent-streams = 100
```

http2-max-frame-size

Use the **http2-max-frame-size** stanza entry to define the maximum size of the body of a single HTTP/2 protocol frame sent over the HTTP/2 network connection.

Syntax

```
http2-max-frame-size = number_of_bytes
```

Description

This stanza entry defines the maximum size of the body of a single HTTP/2 protocol frame sent over the HTTP/2 network connection to a junctioned server.

This configuration item may be customized for a particular junction by adding the adjusted configuration item to a `[junction:{jct_id}]` stanza, where `'{jct-id}'` refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

Options

number_of_bytes
The maximum size of the body of a single HTTP/2 protocol frame that can be sent over the HTTP/2 network connection.

Usage

This stanza entry is optional.

Default value

None.

Example

```
http2-max-frame-size = 16384
```

http2-max-header-list-size

Use the **http2-max-header-list-size** stanza entry to define the maximum size of headers that can be sent in a request on an HTTP/2 stream.

Syntax

```
http2-max-header-list-size = number_of_bytes
```

Description

This stanza entry defines the maximum size of headers in bytes that can be sent in a request on an HTTP/2 stream to a junctioned server. A value of -1 denotes the unlimited setting and is not recommended in a production WebSEAL environment as memory use in WebSEAL would be unbounded. If this entry is not set, it will default to the value of **[server] max-client-read**.

This configuration item may be customized for a particular junction by adding the adjusted configuration item to a `[junction:{jct_id}]` stanza, where `'{jct-id}'` refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

Options

number_of_bytes
The maximum size of headers that can be sent in a request on an HTTP/2 stream.

Usage

This stanza entry is optional.

Default value

The value of the **max-client-read** entry in the **[server]** stanza.

Example

```
http2-max-header-list-size = 32768
```

http-header-attributes

Use the `http-header-attributes` stanza entry to define the credential attributes which will be added as HTTP headers to the request.

Syntax

```
http-header-attributes = <attr-name>{:::<hdr-name>}
```

Description

Specifies which attributes from the credential will be inserted, as HTTP headers, into requests which are sent to the junctioned server.

This entry may be repeated multiple times, once for each attribute which is to be added.

This configuration item may be customized for a particular junction by adding the adjusted configuration item to a `[junction:{jct_id}]` stanza, where '`{jct-id}`' refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

Options

<attr-name>
The name of the credential attribute.

<hdr-name>
The name of the HTTP header which will contain the attribute. If the header name is not specified, the name of the attribute will be used as the HTTP header name.

Usage

This stanza entry is optional.

Default Value

None.

Example

```
http-header-attributes = AZN_CRED_PRINCIPAL_NAME:::principal
```

http-timeout

Syntax

```
http-timeout = number_of_seconds
```

Description

Integer value indicating the timeout, in seconds, for sending to and reading from a TCP junction.

You can customize this configuration item for a particular junction by adding the adjusted configuration item to a **[junction:*{junction_name}*]** stanza.

where *{junction_name}* refers to the junction point for a standard junction (including the leading / character) or the virtual host label for a virtual host junction.

Options

number_of_seconds
Integer value indicating the timeout, in seconds, for sending to and reading from a TCP junction. The minimum value is 0. When the value is 0, there is no timeout. WebSEAL does not impose a maximum value.

Usage

This stanza entry is required.

Default value

120

```
http-timeout = 120
```

https-timeout

Syntax

```
https-timeout = number_of_seconds
```

Description

Integer value indicating the timeout, in seconds, for sending to and reading from a Secure Socket Layer (SSL) junction.

You can customize this configuration item for a particular junction by adding the adjusted configuration item to a **[junction:*{junction_name}*]** stanza.

where *{junction_name}* refers to the junction point for a standard junction (including the leading / character) or the virtual host label for a virtual host junction.

Options

number_of_seconds
Integer value indicating the timeout, in seconds, for sending to and reading from a Secure Socket Layer (SSL) junction. The minimum value is 0. When the value is 0, there is no timeout. WebSEAL does not impose a maximum value.

Usage

This stanza entry is required.

Default value

120

```
https-timeout = 120
```

ignore-svc-unavailable

Use **ignore-svc-unavailable** to control whether WebSEAL handles a 503 'Service Unavailable' from a back-end server or returns it to the client.

Syntax

```
ignore-svc-unavailable = {true|false}
```

Description

The following configuration entry sets whether WebSEAL should handle a 503 "Service Unavailable" from a back-end server or return it to the client.

This configuration item might be customized for a particular junction by adding the adjusted configuration item to a **[junction:*{jct_id}*]** stanza, where *{jct-id}* refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

Type

Boolean

Usage

This stanza entry is optional.

Default Value

False

Example

```
ignore-svc-unavailable = false
```

kerberos-principal-name

Use the **kerberos-principal-name** entry to set the service principal name of the impersonating user when creating a Kerberos token.

Syntax

```
kerberos-principal-name = principal_name
```

Description

The service principal name can be determined by executing the Microsoft utility **setspn**. For example:

```
setspn -L user
```

where *user* is the identity of the WebSEAL account.

This configuration item may be customized for a particular junction by adding the adjusted configuration item to a `[junction:{jct_id}]` stanza, where *'{jct-id}'* refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

Options

principal_name
The service principal name of the impersonating user when creating a Kerberos token.

Usage

This stanza entry is required when Kerberos SSO authentication for junctions is enabled.

Default value

None.

Example

```
kerberos-principal-name = HTTP/webseal@<realm>
```

kerberos-service-name

Use the **kerberos-service-name** entry to set the service principal name of the target.

Syntax

```
kerberos-service-name = service-name
```

Description

The service principal name can be determined by executing the Microsoft utility **setspn** (that is, `setspn -L user`, where `user` is the identity of the back-end Web server's account). This configuration item may be customized for a particular junction by adding the adjusted configuration item to a `[junction:{jct_id}]` stanza, where `'{jct-id}'` refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

The format is:

```
kerberos-service-name = HTTP/<username>@<realm>
```

Options

service-name
The service principal name of the target.

Usage

This stanza entry is required when Kerberos SSO authentication for junctions is enabled.

Default value

None.

Example

```
kerberos-service-name = HTTP/myservice@<realm>
```

kerberos-sso-enable

Use the **kerberos-sso-enable** entry to enable or disable SSO for junctions.

Syntax

```
kerberos-sso-enable = {yes|true|no|false}
```

Description

You can customize this configuration item for a particular junction by adding the adjusted configuration item to a **[junction:{junction_name}]** stanza.

Options

yes
Enable.

true
Enable.

no
Disable.

false
Disable.

Usage

This stanza entry is required.

Default value

no.

Example

```
kerberos-sso-enable = no
```

kerberos-user-identity

Use the **kerberos-user-identity** stanza entry to enable and define a custom user principal name (UPN). The custom UPN can be constructed from either plain text or the contents of credential attributes.

Syntax

```
kerberos-user-identity = username@domain
kerberos-user-identity = username
kerberos-user-identity = @domain
kerberos-user-identity = fqdn
```

Description

An administrator can overwrite the UPN or sections of the UPN for Kerberos constrained delegation users with this entry. The replacement information can be either plain text or names of credential attributes that store the required information. If you specify plain text, the text is directly copied into the UPN sections. If you specify names of credential attributes, the replacement text is fetched from the value of the corresponding credential attribute.

The domain information can also be extracted from the DC elements of the user's DN through the attribute **attr:dn**.

If no user name is defined, the client credential name is used.

If no domain is defined, the WebSEAL service account domain is used.

The domain value must be uppercase. Any input data that is not uppercase is automatically converted to uppercase. The domain must also be added as a realm to the Kerberos configuration.

Options

username@domain

Replaces both the user name and the domain separately.

username

Replaces only the user name. The WebSEAL service account domain is used as the user domain.

@domain

Replaces only the domain. The user name is obtained from the client credential.

fqdn

Replaces both the user name and domain with a single attribute. The value of this attribute must contain both the user name and the domain.

Usage

This stanza entry is optional. It can be customized for a particular junction in the **[junction:** *junction_name***]** stanza.

Default value

None

Example

```
kerberos-user-identity = bob@IBM.COM
kerberos-user-identity = attr:SamAccountName@IBM.COM
kerberos-user-identity = @attr:dn
kerberos-user-identity = attr:FQDN
```

managed-cookies-list

Syntax

```
managed-cookies-list = list
```

Description

The managed-cookies-list contains a comma-separated list of patterns that will be matched against the names of cookies returned by junctioned servers. Cookies with names that match the patterns in this list are stored in the WebSEAL cookie jar and not returned to the client. Cookies that do not match these patterns are returned to the client browser.

The WebSEAL cookie jar is turned off by not specifying any cookies in the **managed-cookies-list**.

You can customize this configuration item for a particular junction by adding the adjusted configuration item to a **[junction:*{junction_name}*]** stanza.

where *{junction_name}* refers to the junction point for a standard junction (including the leading / character) or the virtual host label for a virtual host junction.

Options

list
A comma-separated list of pattern-matched cookie names.

Usage

This stanza entry is optional.

Default value

This option is empty by default.

```
managed-cookies-list = JSESS*,Ltpa*
```

match-vhj-first

Helps determine the order in which WebSEAL searches for a request in a standard or a virtual host junction table.

Syntax

```
match-vhj-first = {yes|no}
```

Description

WebSEAL manages separate junction tables for standard and virtual host junctions. When a request comes in, WebSEAL searches the virtual host junction table first. If WebSEAL does not find a match, it searches the table that manages standard junctions. The **match-vhj-first** configuration can reverse the search order so that WebSEAL searches the standard junction table before searching the virtual host junction table.

You can customize this configuration item for a particular junction by adding the adjusted configuration item to a **[junction:*{junction_name}*]** stanza, where *{junction_name}* refers to the junction point for a standard junction (including the leading / character) or the virtual host label for a virtual host junction.

Options

yes
WebSEAL searches the virtual host junction table first.

no
WebSEAL searches the standard junction table first.

Usage

This stanza entry is not optional.

Default value

yes

Example

The following example tells WebSEAL to search the standard junction table first:

```
match-vhj-first = no
```

max-cached-persistent-connections

Syntax

```
max-cached-persistent-connections = number_of_connections
```

Description

The maximum number of persistent connections that will be stored in the cache for future use. Connections with junctioned Web servers will be cached for future use unless the configured limit (as defined by this configuration entry) is reached, or unless the **connection:close** header is received in the HTTP response.

Note: If this setting is enabled, there is the potential for different user sessions to use the same connection when processing junction requests. To disable the persistent connection functionality, specify a **max-cached-persistent-connections** value of zero (`0`).

You can customize this configuration item for a particular junction by adding the adjusted configuration item to a **[junction:*{junction_name}*]** stanza.

where *{junction_name}* refers to the junction point for a standard junction (including the leading / character) or the virtual host label for a virtual host junction.

Options

number_of_connections
Integer value indicating the maximum number of persistent connections that will be stored in the cache for future use. A value of zero (0) disables this support. WebSEAL imposes no maximum on this value.

Usage

This stanza entry is required.

Default value

0

```
max-cached-persistent-connections = 0
```

max-jct-read

Use the **max-jct-read** stanza entry to control the amount of header data WebSEAL will read from responses.

Syntax

```
max-jct-read = number_of_bytes
```

Description

Maximum size, in bytes, of headers WebSEAL read from responses. By default, WebSEAL read headers up to 65536 bytes in length. When larger headers are expected, for example in the case of an EAI authentication where the user belongs to many groups, this parameter must be increased in order for WebSEAL to parse the complete header.

You can customize this configuration item for a particular junction by adding the adjusted configuration item to a **[junction:{junction_name}]** stanza, where *{junction_name}* refers to the junction point for a standard junction (including the leading /character) or the virtual host label for a virtual host junction.

Options

number_of_bytes
The maximum number of bytes of header data WebSEAL will read from responses.

Usage

This stanza entry is optional.

Default value

65536

Example

```
max-jct-read = 131072
```

persistent-con-timeout

Syntax

```
persistent-con-timeout = number_of_seconds
```

Description

Indicates the maximum number of seconds a persistent connection can remain idle in a cache before the connection is cleaned up and closed by WebSEAL.

Use an integer value lower than the configured maximum connection lifetime for the junctioned web server. For example, the connection lifetime for a junctioned Apache web server is controlled by the **KeepAliveTimeout** configuration entry.

You can customize the **persistent-con-timeout** configuration item for a particular junction by adding the adjusted configuration item to a **[junction:*{junction_id}*]** stanza.

where *{junction_id}* refers to the junction point for a standard junction (including the leading / character) or the virtual host label for a virtual host junction.

Note: If you do not use an integer value lower than the connection lifetime on the junctioned web server, you might encounter the following problem.

If the **[junction] max-cached-persistent-connections** configuration entry is set to a value greater than zero, WebSEAL reuses its TCP/IP session with the junctioned back-end server. If the junctioned back-end server closes the socket at the same time that WebSEAL starts to use this session to send a request, the request fails.

To send the request again, WebSEAL opens a new TCP/IP session. If the request body is larger than the size that WebSEAL can cache, WebSEAL fails to resend the request and generates a 500 error.

Options

number_of_seconds
: Integer value that indicates the maximum number of seconds a persistent connection can remain idle in a cache before the connection is closed by WebSEAL. The minimum value is 1. WebSEAL does not impose a maximum value.

Usage

This stanza entry is required.

Default value

5

Example

```
persistent-con-timeout = 5
```

ping-method

Syntax

```
ping-method = method
```

Description

The WebSEAL server performs a periodic background ping of each junctioned Web server, to determine whether it is running. The optional **ping-method** entry sets the HTTP request type used in these pings. The valid options include any valid HTTP request method (for example, *HEAD* or *GET*, for HTTP HEAD and HTTP GET requests respectively).

This configuration item may be customized for a particular junction by adding the adjusted configuration item to a **[junction:*{junction_name}*]** stanza.

where *{junction_name}* refers to the junction point for a standard junction (including the leading / character) or the virtual host label for a virtual host junction.

Options

method
Perform a HTTP request using the specified method to determine the state of the junctioned server.

Usage

This stanza entry is optional.

Default value

HEAD

Example

```
ping-method = GET
```

ping-response-code-rules

Use the **ping-response-code-rules** configuration entry to define the rules that are used to determine whether the HTTP status code of the ping responses indicate a healthy or an unhealthy junctioned Web server.

Syntax

```
ping-response-code-rules = list
```

Description

The WebSEAL server performs a periodic background ping of each junctioned Web server, to determine whether the junctioned Web server is running. The optional **ping-response-code-rules** configuration entry defines the rules that are used to determine whether the HTTP status code of the ping responses indicate a healthy or an unhealthy junctioned Web server.

If valid values are configured for both **ping-response-code-rules** and **response-code-rules**, the specified **ping-response-code-rules** will be applied to the ping requests initiated by WebSEAL, and other requests will be matched against **response-code-rules** to determine the server state.

If a valid **ping-response-code-rules** value is configured but **response-code-rules** is not, the specified **ping-response-code-rules** will be applied to the ping requests initiated by WebSEAL, and other requests will not be used to determine the server state. In this case, **ping-response-code-rules** are the only rules used to determine the server state.

If the **ping-response-code-rules** configuration entry is not set, the rules that are specified by the **response-code-rules** configuration entry will also apply to ping requests.

If the **ping-attempt-threshold** entry is configured, when the junction is marked as running, it must fail this number of consecutive ping requests before it is marked as not running. If this entry is not set it is default to 1.

If the **recovery-ping-attempt-threshold** entry is configured, when the junction is marked as not running, it must return this number of consecutive successful recovery ping responses before it is marked as running. If this entry is not set it is default to 1.

The configuration entry contains a space separated list of rules. Each rule has the format: `[+|-]<code>` (e.g. `-50?`)

where:

+
: Indicates that this is a healthy response code.

-
: Indicates that this is an unhealthy response code.

<code>
: The corresponding response code, which can also contain pattern matching characters such as `*` and `?`

The HTTP response codes are evaluated against each rule in sequence until a match is found. The corresponding code (+ | -) determines whether the junctioned Web server is healthy or not.If the response code matches no configured rules, the junctioned Web server is considered healthy.

You can customize this configuration item for a particular junction by adding the adjusted configuration item to a **[junction:*{junction_name}*]** stanza, where *{junction_name}* refers to the junction point for a standard junction (including the leading / character) or the virtual host label for a virtual host junction.

Options

list
: A space separated list of response code rules. These rules determine whether the ping response from a junctioned Web server indicates a healthy or an unhealthy server.

Usage

This stanza entry is optional.

Default value

None.

Example

```
ping-response-code-rules = +2?? -*
```

ping-attempt-threshold

Use this entry to define the number of consecutive failed ping requests before the junctioned server will be marked as not running.

Syntax

```
ping-attempt-threshold = number
```

Description

The WebSEAL server performs a periodic background ping of each junctioned Web server, to determine whether it is running.

If this entry is configured, when the junction is marked as running, it must fail this number of consecutive ping requests before it will be marked as not running.

If this entry is not set it will default to 1.

You can customize this configuration item for a particular junction by adding the adjusted configuration item to a **[junction:*{junction_name}*]** stanza, where *{junction_name}* refers to the junction point for a standard junction (including the leading / character) or the virtual host label for a virtual host junction.

Options

number
The number of consecutive failed ping requests before it will be marked as not running.

Usage

This stanza entry is optional.

Default value

1

Example

```
ping-attempt-threshold = 1
```

ping-time

Syntax

```
ping-time = number_of_seconds
```

Description

Integer value indicating the number of seconds between pings issued by the WebSEAL server. The pings are issued periodically in the background to verify that junctioned WebSEAL servers are running.

If the server is deemed not running, the **recovery-ping-time** value determines the interval at which pings are sent until the server is running. The type of ping used is determined by the **ping-method** value. HTTP response code rules can be defined using the **response-code-rules** configuration entry.

You can customize this configuration item for a particular junction by adding the adjusted configuration item to a **[junction:*{junction_name}*]** stanza, where *{junction_name}* refers to the junction point for a standard junction (including the leading / character) or the virtual host label for a virtual host junction.

Options

number_of_seconds
Integer value indicating the number of seconds between pings issued by the WebSEAL server. The minimum value is 1. WebSEAL does not impose a maximum value.

Usage

To turn this ping off, set this entry to zero. If this entry is set to zero, the **recovery-ping-time** must be set.

Default value

300

Example

```
ping-time = 300
```

ping-timeout

Use this entry to set a different timeout value for the 'ping' operations.

Syntax

```
ping-timeout = timeout
```

Description

Timeout (in seconds) for sending ping requests to, and reading ping responses from, the junction. The value must be an integer greater than or equal to zero. A value of zero causes WebSEAL to wait indefinitely. If no value is set the standard junction timeout values apply to the ping requests. This configuration item might be customized for a particular junction by adding the adjusted configuration item to a `[junction:{jct_id}]` stanza, where `'{jct-id}'` refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

If this entry is not set, it defaults to the value of the `http-timeout` or `https-timeout` configuration entries, depending on the type of junction that is being accessed.

Options

timeout
Timeout (in seconds) for sending ping requests to, and reading ping responses from, the junction

Usage
This stanza entry is optional.

Default value
The value defaults to the value of the `http-timeout` or `https-timeout` configuration entries, depending on the type of junction that is being accessed.

Example

```
ping-timeout = 30
```

ping-uri

Syntax

```
ping-uri = uri
```

Description

The WebSEAL server performs a periodic background ping of each junctioned Web server to determine whether it is running. The optional **ping-uri** configuration entry defines the URI that is accessed by the ping request. The defined URI is relative to the root Web space of the junctioned Web server. If the URI is missing, this value defaults to a `/`.

You can customize this configuration item for a particular junction by adding the adjusted configuration item to a **[junction:*{junction_name}*]** stanza.

where *{junction_name}* refers to the junction point for a standard junction (including the leading / character) or the virtual host label for a virtual host junction.

Options

uri
The URI that is accessed by the ping request.

Usage

This stanza entry is optional.

Default value

/

```
ping-uri = /apps/status
```

recovery-ping-time

Syntax

```
recovery-ping-time = number_of_seconds
```

Description

The WebSEAL server performs a periodic background ping of each junctioned Web server, to determine whether it is running. This entry sets the interval, in seconds, between pings when the server is determined to be not running.

You can customize this configuration item for a particular junction by adding the adjusted configuration item to a **[junction:*{junction_name}*]** stanza, where *{junction_name}* refers to the junction point for a standard junction (including the leading / character) or the virtual host label for a virtual host junction.

Options

number_of_seconds
Integer value indicating the number of seconds between pings issued by the WebSEAL server to a junctioned server that is determined to be not running. The minimum value is 1. WebSEAL does not impose a maximum value.

Usage

If this entry is not set, the **recovery-ping-time** defaults to the **ping-time** value.

Default value

300

Example

```
recovery-ping-time = 300
```

recovery-ping-attempt-threshold

Use this entry to define the number of consecutive successful recovery ping responses before a stopped junctioned server will be marked as running.

Syntax

```
recovery-ping-attempt-threshold = number
```

Description

The WebSEAL server performs a periodic background ping of each junctioned Web server, to determine whether it is running.

If this entry is configured, when the junction is marked as not running, it must return this number of consecutive successful recovery ping responses before it will be marked as running.

If this entry is not set it will default to 1.

You can customize this configuration item for a particular junction by adding the adjusted configuration item to a **[junction:*{junction_name}*]** stanza, where *{junction_name}* refers to the junction point for a standard junction (including the leading / character) or the virtual host label for a virtual host junction.

Options

number
The number of consecutive successful recovery ping responses before it will be marked as running.

Usage

This stanza entry is optional.

Default value

1

Example

```
recovery-ping-attempt-threshold = 1
```

reset-cookies-list

Syntax

```
reset-cookies-list = list
```

Description

Determines which cookies are reset when the user session is logged out. The request received from the client and the response sent back to the client are both examined for matching cookies.

You can customize this configuration item for a particular junction by adding the adjusted configuration item to a **[junction:*{junction_name}*]** stanza.

where *{junction_name}* refers to the junction point for a standard junction (including the leading / character) or the virtual host label for a virtual host junction.

Options

list

A comma-separated list of patterns. WebSEAL will reset any cookies with names that match the patterns in this list.

Usage

This stanza entry is required.

Default value

None.

```
reset-cookies-list = JSESS*,Ltpa*
```

response-code-rules

When a response of a client-initiated request is returned from the junctioned server, the optional **response-code-rules** configuration entry defines the rules that are used to determine from the HTTP status code of the responses whether the junctioned Web server is in a healthy or an unhealthy state.

Syntax

```
response-code-rules = list
```

Description

The optional **response-code-rules** configuration entry defines the rules that are used to determine whether HTTP responses indicate a healthy or an unhealthy junctioned Web server.

This configuration entry will apply to all requests if the **ping-response-code-rules** configuration entry has not been set. Otherwise, it will only apply to all client-initiated requests.

If this configuration entry is empty, and a **ping-response-code-rules** configuration has been specified, the response code received from client-initiated requests will not impact the health of the junction.

The configuration entry contains a space separated list of rules. Each rule has the format: [+|-]<code> (e.g. -50?)

where:

+

Indicates that this is a healthy response code.

-

Indicates that this is an unhealthy response code.

<code>

The corresponding response code, which can also contain pattern matching characters such as * and ?

The HTTP response codes are evaluated against each rule in sequence until a match is found. The corresponding code (+|-) determines whether the junctioned Web server is healthy or not. If the response code matches no configured rules, the junctioned Web server is considered healthy.

You can customize this configuration item for a particular junction by adding the adjusted configuration item to a **[junction:*{junction_name}*]** stanza, where *{junction_name}* refers to the junction point for a standard junction (including the leading / character) or the virtual host label for a virtual host junction.

Options

list
A space separated list of response code rules. These rules determine whether the response from a junctioned Web server indicates a healthy or an unhealthy server.

Usage

This stanza entry is optional.

Default value

None.

Example

```
response-code-rules = +2?? -*
```

server-hostname-validation

Use the `server-hostname-validation` stanza entry to control whether WebSEAL performs hostname validation on server certificates presented by Junctioned servers.

Syntax

```
server-hostname-validation = {disabled|critical|warning}
```

Description

Specifies whether hostname validation will be performed on the server certificates which are presented by junctioned servers. If enabled, the DNS hostname of the configured server will be checked against the CN and SAN fields of the server certificate.

If the expected CN is specified, using the '-0' option, during junction creation the hostname validation will not be performed.

You can customize this configuration item for a particular junction by adding the adjusted configuration item to a `[junction:{junction_name}]` stanza, where *{junction_name}* refers to the junction point for a standard junction (including the leading / character) or the virtual host label for a virtual host junction.

Options

`disabled`
No hostname validation will take place. This is the default.

`critical`
Hostname validation will take place and connections will be rejected if the validation fails.

`warning`
Hostname validation will take place, but connections will still be allowed if the validation fails. A warning message will however be displayed.

Usage

This stanza entry is optional.

Default value

`default`

Example

```
server-hostname-validation = critical
```

support-virtual-host-domain-cookies

Syntax

```
support-virtual-host-domain-cookies = {yes|no}
```

Description

If **allow-backend-domain-cookies** is set to `yes`, then this option modifies how WebSEAL validates the domain. This option has no effect if `validate-backend-domain-cookies = no`.

You can customize this configuration item for a particular junction by adding the adjusted configuration item to a **[junction:*{junction_name}*]** stanza.

where *{junction_name}* refers to the junction point for a standard junction (including the leading / character) or the virtual host label for a virtual host junction.

Options

yes
If set to "yes" then the domain cookie is validated by comparing it with the virtual host specified for a backend server with the **-v** junction option.

no
If set to "no", or if no virtual host was specified for a junction, then the fully qualified host name is compared with the domain value of a backend cookie for validation.

Usage

This stanza entry is required.

Default value

`yes`

```
support-virtual-host-domain-cookies = yes
```

use-new-stateful-on-error

Syntax

```
use-new-stateful-on-error = {yes|no}
```

Description

Control how WebSEAL responds to a stateful server that becomes unavailable.

This configuration item may be customized for a particular junction by adding the adjusted configuration item to a `[junction:`*`{junction_name}`*`]` stanza.

where *{junction_name}* refers to the junction point for a standard junction (including the leading / character) or the virtual host label for a virtual host junction. For example:

```
[junction:/WebApp]
```

Options

yes
When set to "yes" and the original server becomes unavailable during a session, WebSEAL directs the user's next request (containing the original stateful cookie) to a new replica server on the same stateful junction. If a new replica server is found on that stateful junction, and is responsive to the request, WebSEAL sets a new stateful cookie on the user's browser. Subsequent requests during this same session (and containing the new stateful cookie) are directed to this same new server.

no
When set to "no" and the original server becomes unavailable during a session, WebSEAL does not direct the user's subsequent requests to a new replica server on the same stateful junction. Instead, WebSEAL returns an error and attempts to access the same server for subsequent requests by the user during this session.

Usage

This stanza entry is required.

Default value

no

Example

```
use-new-stateful-on-error = yes
```

validate-backend-domain-cookies

Syntax

```
validate-backend-domain-cookies = {yes|no}
```

Description

Specifies how WebSEAL validates the domain.

You can customize this configuration item for a particular junction by adding the adjusted configuration item to a **[junction:*{junction_name}*]** stanza.

where *{junction_name}* refers to the junction point for a standard junction (including the leading / character) or the virtual host label for a virtual host junction.

Options

yes
If set to "yes" then domain cookies that adhere to the cookie specification are forwarded to the user. If the fully qualified host name of the originating back-end machine is the domain, then the cookie is forwarded to the user with no domain specified.

no
If set to "no", then all domain cookies are forwarded to the user, regardless of their content.

Usage

This stanza entry is required.

Default value

yes

```
validate-backend-domain-cookies = yes
```

[junction:*junction_name*] stanza

Note: This stanza is optional and must be manually inserted into the WebSEAL configuration file. The *junction_name* in the stanza name is the junction point for a standard junction (including the leading / character) or the virtual host label for a virtual host junction. For details about the configuration entries supported in this junction specific stanza, see the description of the corresponding configuration entry in the **[junction]** stanza.

[jwt]

The JWT stanza is used to control the generation of JSON Web Tokens. It contains the default JWT configuration entries - which are used when the corresponding configuration entry is not present in the junction-specific stanza (`[jwt:<jct-id>]`). The `applies-to` configuration entry is used to determine which junctions use this stanza.

applies-to

The `applies-to` configuration entry specifies the rules that are used to determine which junctions use the configuration that is located in this stanza as default configuration.

Syntax

```
applies-to =[+|-]<jct-id>
```

Description

The rules that are used to determine which junctions use the configuration that is located in this stanza as default configuration. This entry can be repeated multiple times, once for each rule that is to be defined.

The junction identifier is evaluated against each rule in sequence until a match is found. The corresponding code (+|-) is then used to determine whether the configuration is used as a default value by the junction. If the junction identifier matches no configured rules, the stanza is used by the junction.

Options

+
Indicates that the configuration is used by the junction.

-
Indicates that the configuration is not used by the junction.

<jct-id>
The identifier of the junction, either the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction. This field can contain the '*?' pattern matching characters.

Usage

This stanza entry is optional.

Default Value

None

Example

```
applies-to = +/junction_*
applies-to = -*
```

claim

Use the claim stanza entry to define an attribute that is added to the JWT as a claim.

Syntax

```
claim = [text|attr]{.<type>}{.array}::<value>{::<claim-name>}
```

Description

A claim that is added to the generated JWT. The configuration entry can be specified multiple times for each claim that is added to the JWT.

Options

text
Used to indicate that literal text is added as a claim.

attr
Used to indicate that the claim is obtained from a credential attribute.

<type>

The source can be qualified with a 'type' (delimited by a dot). The valid types include:

bool
The value is added as a boolean.

int
The value is added as an integer.

string
The value is added as a string.

If no type is specified, values are added as strings.

array

If specified, the value is added as an array, regardless of how many values are present. If not specified, any single value attribute is added as a single element and any multi-valued attributes as an array.

<value>

For text

The value that is added.

An array of values can be specified by surrounding the string with square brackets ([]). A comma is used to delimit each individual value. The comma can be escaped with a backslash character if a literal comma is required in the value.

For attr

The name of a credential attribute. The '*' and '?' pattern matching characters can be used to match multiple attributes. Pattern matching characters are ignored if the '<claim-name>' is specified.

<claim-name>
The name of the claim to be added to the JWT. Nested objects can be specified, separating the name of each object field with a . (dot). If the name of a field itself embeds a dot, it must be escaped with a backslash character (for example \.).

Usage

This stanza entry is required when the reverse proxy is generating a JWT that is to be sent to a junctioned server.

Default value

None

Example

```
claim = text::www.ibm.com::iss
claim = attr.int::BUSINESS_PHONE_NUMBER::phone.business
claim = attr::AZN_CRED_PRINCIPAL_NAME::sub
claim = attr.array::AZN_CRED_GROUPS::groupList
claim = attr.int.array::postCodes::postCodesList
claim = attr::AZN_*
claim = text.bool::true::is_jwt
```

hdr-format

Use the `hdr-format` stanza entry to define the format of the HTTP header which will contain the JWT.

Syntax

```
hdr-format = <format>
```

Description

The format of the HTTP header which will contain the generated JWT. The %TOKEN% string will be substituted with the value of the generated JWT.

Options

<format>
The format of the header which will contain the generated JWT.

Usage

This stanza entry is optional.

Default Value

```
%TOKEN%
```

Example

```
hdr-format = Bearer %TOKEN%
```

hdr-name

Use the `hdr-name` stanza entry to specify the name of the HTTP header which will contain the JWT.

Syntax

```
hdr-name = <header name>
```

Description

The name of the HTTP header which will contain the generated JWT.

Options

<header name>
The name of the header which will contain the generated JWT.

Usage

This stanza entry is required when generating a JWT which is to be sent to a junctioned server.

Default value

None

Example

```
hdr-name = iv-jwt
```

include-empty-claims

Use `include-empty-claims` to control whether empty claims are included in the JWT for missing credential attributes.

Syntax

```
include-empty-claims = {yes|true|no|false}
```

Description

Claims that correspond to attributes that are not present in the credential are not added to the JWT. Use this configuration entry to specify that missing attributes are still added as an empty string.

Options

yes
Empty claims are included.

true
Empty claims are included.

no
Empty claims are not included.

false
Empty claims are not included.

Usage

This stanza entry is optional.

Default Value

`false`

Example

```
include-empty-claims = true
```

key-label

Use the `key-label` stanza entry to specify the name of the key which is used to sign the JWT.

Syntax

```
key-label = <key label>
```

Description

The label associated with the server key which is used to sign the JWT. This key **must** exist in the key file which is used to secure junction communication(for example, defined by the '`jct-cert-keyfile`' or '`webseal-cert-keyfile`' configuration entries).

Options

<key-label>
The label of the key which is used to sign the JWT.

Usage

This stanza entry is required when generating a JWT which is to be sent to a junctioned server.

Default Value

None

Example

```
key-label = jwt_key
```

lifetime

Use the `lifetime` stanza entry to define the lifetime of a generated JWT.

Syntax

```
lifetime = <lifetime>
```

Description

The length of time, in seconds, that a JWT will remain valid. A new JWT will be automatically created when the current JWT expires. A value of 0 indicates that the expiry time will be set to match the expiry time of the session.

Options

<lifetime>
The lifetime, in seconds, of the generated JWT.

Usage

This stanza entry is optional.

Default Value

```
0
```

Example

```
lifetime = 600
```

renewal-window

Use the `renewal-window` stanza entry to define by how much the lifetime of a generated JWT will be reduced.

Syntax

```
renewal-window = <seconds>
```

Description

The length of time, in seconds, by which the expiry time of a JWT will be reduced. This entry is used to make allowances for differences in system times and transmission times for the JWT.

Options

<seconds>
The length of time, in seconds, by which the lifetime of a generated JWT will be reduced.

Usage

This stanza entry is optional.

Default Value

15

Example

```
renewal-window = 30
```

[jwt:<jct-id>]

The JWT stanza is used to control the generation of JSON Web Tokens for the specified junction. The '`{jct-id}`' refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

claim

Use the claim stanza entry to define an attribute that is added to the JWT as a claim.

Syntax

```
claim = [text|attr]{.<type>}{.array}::<value>{::<claim-name>}
```

Description

A claim that is added to the generated JWT. The configuration entry can be specified multiple times for each claim that is added to the JWT.

Options

text
Used to indicate that literal text is added as a claim.

attr
Used to indicate that the claim is obtained from a credential attribute.

<type>

The source can be qualified with a 'type' (delimited by a dot). The valid types include:

bool
The value is added as a boolean.

int
The value is added as an integer.

string
The value is added as a string.

If no type is specified, values are added as strings.

array

If specified, the value is added as an array, regardless of how many values are present. If not specified, any single value attribute is added as a single element and any multi-valued attributes as an array.

<value>

For text

The value that is added.

An array of values can be specified by surrounding the string with square brackets ([]). A comma is used to delimit each individual value. The comma can be escaped with a backslash character if a literal comma is required in the value.

For attr

The name of a credential attribute. The '*' and '?' pattern matching characters can be used to match multiple attributes. Pattern matching characters are ignored if the '<claim-name>' is specified.

<claim-name>
The name of the claim to be added to the JWT. Nested objects can be specified, separating the name of each object field with a . (dot). If the name of a field itself embeds a dot, it must be escaped with a backslash character (for example \.).

Usage

This stanza entry is required when the reverse proxy is generating a JWT that is to be sent to a junctioned server.

Default value

None

Example

```
claim = text::www.ibm.com::iss
claim = attr.int::BUSINESS_PHONE_NUMBER::phone.business
claim = attr::AZN_CRED_PRINCIPAL_NAME::sub
claim = attr.array::AZN_CRED_GROUPS::groupList
claim = attr.int.array::postCodes::postCodesList
claim = attr::AZN_*
claim = text.bool::true::is_jwt
```

hdr-format

Use the `hdr-format` stanza entry to define the format of the HTTP header which will contain the JWT.

Syntax

```
hdr-format = <format>
```

Description

The format of the HTTP header which will contain the generated JWT. The %TOKEN% string will be substituted with the value of the generated JWT.

Options

<format>
The format of the header which will contain the generated JWT.

Usage

This stanza entry is optional.

Default Value

```
%TOKEN%
```

Example

```
hdr-format = Bearer %TOKEN%
```

hdr-name

Use the `hdr-name` stanza entry to specify the name of the HTTP header which will contain the JWT.

Syntax

```
hdr-name = <header name>
```

Description

The name of the HTTP header which will contain the generated JWT.

Options

<header name>
The name of the header which will contain the generated JWT.

Usage

This stanza entry is required when generating a JWT which is to be sent to a junctioned server.

Default value

None

Example

```
hdr-name = iv-jwt
```

include-empty-claims

Use `include-empty-claims` to control whether empty claims are included in the JWT for missing credential attributes.

Syntax

```
include-empty-claims = {yes|true|no|false}
```

Description

Claims that correspond to attributes that are not present in the credential are not added to the JWT. Use this configuration entry to specify that missing attributes are still added as an empty string.

Options

yes
Empty claims are included.

true
Empty claims are included.

no
Empty claims are not included.

false
Empty claims are not included.

Usage

This stanza entry is optional.

Default Value

`false`

Example

```
include-empty-claims = true
```

key-label

Use the `key-label` stanza entry to specify the name of the key which is used to sign the JWT.

Syntax

```
key-label = <key label>
```

Description

The label associated with the server key which is used to sign the JWT. This key **must** exist in the key file which is used to secure junction communication(for example, defined by the '`jct-cert-keyfile`' or '`webseal-cert-keyfile`' configuration entries).

Options

<key-label>
The label of the key which is used to sign the JWT.

Usage

This stanza entry is required when generating a JWT which is to be sent to a junctioned server.

Default Value

None

Example

```
key-label = jwt_key
```

lifetime

Use the `lifetime` stanza entry to define the lifetime of a generated JWT.

Syntax

```
lifetime = <lifetime>
```

Description

The length of time, in seconds, that a JWT will remain valid. A new JWT will be automatically created when the current JWT expires. A value of 0 indicates that the expiry time will be set to match the expiry time of the session.

Options

<lifetime>
The lifetime, in seconds, of the generated JWT.

Usage

This stanza entry is optional.

Default Value

```
0
```

Example

```
lifetime = 600
```

renewal-window

Use the `renewal-window` stanza entry to define by how much the lifetime of a generated JWT will be reduced.

Syntax

```
renewal-window = <seconds>
```

Description

The length of time, in seconds, by which the expiry time of a JWT will be reduced. This entry is used to make allowances for differences in system times and transmission times for the JWT.

Options

<seconds>
The length of time, in seconds, by which the lifetime of a generated JWT will be reduced.

Usage

This stanza entry is optional.

Default Value

15

Example

```
renewal-window = 30
```

[ldap] stanza

auth-timeout

Use the **auth-timeout** stanza entry to configure the timeout for authentication operations.

Syntax

```
auth-timeout = {0|number_seconds}
```

Description

Amount of time in seconds that are allowed for authentication operations before the LDAP server is considered to be down. If specified, this value overrides any value of **timeout** for authentication operations.

Note: Do not specify this parameter in the `ldap.conf` server configuration file.

Options

0
No timeout is allowed.

number_seconds
A positive integer that represents the number of seconds allowed for authentication. There is no range limitation for timeout values.

Usage

This stanza entry is optional.

Default value

`0`

Example

```
auth-timeout = 0
```

auth-using-compare

Use the **auth-using-compare** stanza entry to use a password compare operation for authentication.

Syntax

```
auth-using-compare = {yes|true|no|false}
```

Description

Enables or disables authentication by using password comparison. When disabled, an LDAP bind is used for authentication.

For those LDAP servers that allow it, a compare operation might complete faster than a bind operation.

Options

yes|true
A password compare operation is used to authenticate LDAP users.

no|false
A bind operation is used to authenticate LDAP users.

Usage

This stanza entry is optional.

Default value

The default value, when LDAP is enabled, is `yes`.

Example

```
auth-using-compare = yes
```

basic-user-support

Use the **basic-user-support** configuration entry to enables or disables basic user support.

Syntax

```
basic-user-support = {yes|no}
```

Description

This configuration entry enables or disables basic user support.

Options

yes
Enable basic user support.

no
Disable basic user support.

Usage

This stanza entry is required.

Default value

yes

Example

```
basic-user-support = yes
```

basic-user-pwd-policy

If basic user support is enabled, use this entry to specify whether global password policy is enabled for basic users.

Syntax

```
basic-user-pwd-policy = {true|false}
```

Description

Not all global password policies are available to basic users. See the following table for which policies are supported for basic users.

Table 1. Global user policies for basic users

Policy	Available to basic users
account-expiry-date	Yes
disable-time-interval	No
max-concurrent-web-sessions	Yes

Table 1. Global user policies for basic users (continued)	
Policy	**Available to basic users**
max-login-failures	No
max-password-age	No
max-password-repeated-chars	Yes
min-password-alphas	Yes
min-password-length	Yes
min-password-non-alphas	Yes
password-spaces	Yes
tod-access	Yes

This configuration entry enables or disables the global password policies for basic users.

Options

true
Enable global password policy for basic users.

false
Disable global password policy for basic users.

Usage

This stanza entry is optional.

Default value

`true`

Example

```
basic-user-pwd-policy = true
```

cache-enabled

Use the **cache-enabled** stanza entry to control whether LDAP client-side caching is enabled.

Syntax

```
cache-enabled = {yes|true|no|false}
```

Description

Enable and disable LDAP client-side caching.

Options

yes|true
Enable LDAP client-side caching.

no|false
Disable LDAP client-side caching. Anything other than `yes` | `true`, including a blank value, is interpreted as no | `false`.

Usage

This stanza entry is required.

Default value

yes

Example

```
cache-enabled = yes
```

cache-group-expire-time

Use the **cache-group-expire-time** stanza entry to specify the timeout duration in seconds for group entries in the cache.

Syntax

```
cache-group-expire-time = number_of_seconds
```

Description

Specifies the amount of time to elapse before a group entry in the cache is discarded.

This entry is used only when `cache-enabled = {yes|true}`.

Options

number_of_seconds
Specifies the amount of time to elapse before a group entry in the cache is discarded.

Usage

This stanza entry is optional.

Default value

There is no default value, but when not set the default value used is 300 seconds.

Example

```
cache-group-expire-time = 300
```

cache-group-membership

Use the **cache-group-membership** stanza entry to control whether WebSEAL caches group membership information.

Syntax

```
cache-group-membership = {yes|no}
```

Description

Indicates whether group membership information should be cached.

This entry is used only when `cache-enabled = {yes|true}`

Options

yes
Cache group membership information.

no
Do not cache group membership information.

Usage

This stanza entry is optional.

Default value

There is no default value, but when not set the group information is cached.

Example

```
cache-group-membership = yes
```

cache-group-size

Use the **cache-group-size** stanza entry to set the size of the LDAP group cache.

Syntax

```
cache-group-size = number
```

Description

Specifies the number of entries in the LDAP group cache.

This entry is used only when `cache-enabled = {yes|true}`.

Options

number
Specifies the number of entries in the LDAP group cache.

Usage

This stanza entry is optional.

Default value

There is no default value, but when not set the default value used is 64.

Example

```
cache-group-size = 64
```

cache-policy-expire-time

Use the **cache-policy-expire-time** stanza entry to specify the timeout duration for policy entries in the cache.

Syntax

```
cache-policy-expire-time = number_of_seconds
```

Description

Specifies the amount of time to elapse before a policy entry in the cache is discarded.

This entry is used only when `cache-enabled = {yes|true}`.

Options

number_of_seconds
Specifies the amount of time to elapse before a policy entry in the cache is discarded.

Usage

This stanza entry is optional.

Default value

There is no default value, but when not set the default value used is 30 seconds.

Example

```
cache-policy-expire-time = 30
```

cache-policy-size

Use the **cache-policy-size** stanza entry to set the size of the LDAP policy cache.

Syntax

```
cache-policy-size = number
```

Description

Specifies the number of entries in the LDAP policy cache.

This entry is used only when `cache-enabled = {yes|true}`.

Options

number
Specifies the number of entries in the LDAP policy cache.

Usage

This stanza entry is optional

Default value

There is no default value, but when not set the default value used is 20.

Example

```
cache-policy-size = 20
```

cache-return-registry-id

Use the **cache-return-registry-id** stanza entry to control whether to cache the user identity as it is stored in the registry or cache the user identity as it is entered during authentication.

Syntax

```
cache-return-registry-id = {yes|no}
```

Description

Indicates whether to cache the user identity as it is stored in the registry or cache the value as entered during authentication. Ignored if the cache is not enabled. If not set, the default is no.

Options

yes
Cache the user identity as it is stored in the registry.

no
Cache the user identity as it was entered during authentication.

Usage

This stanza entry is optional

Default value

no

Example

```
cache-return-registry-id = no
```

cache-user-expire-time

Use the **cache-user-expire-time** stanza entry to specify the timeout duration in seconds for user entries in the cache.

Syntax

```
cache-user-expire-time = number_of_seconds
```

Description

Specifies the amount of time to elapse before a user entry in the cache is discarded.

This entry is used only when `cache-enabled = {yes|true}`.

Options

number_of_seconds
Specifies the amount of time to elapse before a user entry in the cache is discarded.

Usage

This stanza entry is optional.

Default value

There is no default value, but when not set the default value used is 30 seconds.

Example

```
cache-user-expire-time = 30
```

cache-user-size

Use the **cache-user-size** stanza entry to set the size of the LDAP user cache.

Syntax

```
cache-user-size = number
```

Description

Specifies the number of entries in the LDAP user cache.

This entry is used only when `cache-enabled = {yes|true}`.

Options

number

Specifies the number of entries in the LDAP user cache.

Usage

This stanza entry is optional.

Default value

There is no default value, but when not set the default value used is 256.

Example

```
cache-user-size = 256
```

cache-use-user-cache

Use the **cache-use-user-cache** stanza entry to control whether WebSEAL uses information from the user cache.

Syntax

```
cache-use-user-cache = {yes|no}
```

Description

Indicates whether to use the user cache information or not.

This entry is used only when `cache-enabled = {yes|true}`

Options

yes
Use the user cache information.

no
Do not use the user cache information.

Usage

This stanza entry is optional.

Default value

There is no default value, but when not set the user cache information is used.

Example

```
cache-use-user-cache = yes
```

default-policy-override-support

Use the **default-policy-override-support** stanza entry to control whether the global (default) policy overrides the user level policy during LDAP searches.

Syntax

```
default-policy-override-support = {yes|true|no|false}
```

Description

Indicates whether default policy overrides user level policy during LDAP searches. When this stanza entry is set to yes, only the default policy is checked.

Options

yes|true
User policy support is disabled and only the global (default) policy is checked. This option allows the user policy to be ignored, even when it is specified.

no|false
User policy support is enabled. When a user policy is specified by the administrator, it overrides the global policy.

Usage

This stanza entry is optional.

Default value

By default, the value is not specified during WebSEAL configuration. When the value is not specified, the default behavior is enable user policy support. This is equivalent to setting this stanza entry to no.

Example

```
default-policy-override-support = yes
```

group-membership-search-all-registries

Use the **group-membership-search-all-registries** stanza entry to control whether all federated registries are search for group membership.

Syntax

```
group-membership-search-all-registries = {yes|true|no|false}
```

Description

To determine the groups that a user is a member of, by default, all federated registries are searched. Setting this option to 'no' means that only the registry which the user is located in will be searched for their group membership.

Options

yes|true
Enable the searching of all federated registries for group membership.

no|false
Disable the searching of all federated registries for group membership.

Usage

This stanza entry is optional.

Default value

yes

Example

```
group-membership-search-all-registries = yes
```

group-membership-search-filter

Use the **group-membership-search-filter** entry to specify the LDAP search filter that is used by Security Verify Access to obtain the group membership for a user.

Syntax

```
group-membership-search-filter = ldap search filter
```

Description

Use this configuration file parameter to specify how to locate Security Verify Access group membership for a user in LDAP.

Options

Specifies the LDAP search filter that is used by Security Verify Access to locate group membership for a user in the LDAP directory server. This filter must be a valid LDAP string search filter as described by the Request for Comments (RFC) 2254 document. Any **%dn%** strings found within the filter are replaced with the distinguished name of the user before the search is performed.

Usage

This stanza entry is optional.

Default value

(|(member=%dn%)(uniqueMember=%dn%)'

Example

This example specifies a search for group membership by using the **member** field of objects, with a class of **groupOfNames**, within the LDAP user record.

```
group-membership-search-filter = (&(objectclass=groupOfNames)(member=%dn%))
```

host

Syntax

```
host = host_name
```

Description

Host name of the LDAP server.

Options

host_name
Valid values for *host_name* include any valid IP host name. The *host_name* does not have to be a fully qualified domain name.

Usage

This stanza entry is required.

Default value

The default value is always taken (during WebSEAL initialization) from the corresponding parameter in the **[ldap]** stanza of the `ldap.conf` configuration file for the LDAP server.

Example

```
host = diamond
host = diamond.example.com
```

login-failures-persistent

Syntax

```
login-failures-persistent = {yes|true|no|false}
```

Description

When set to "yes", login hits are tracked in the registry instead of only in the local process cache.

Persistent login hit recording impacts performance but allows consistent login hit counting across multiple servers.

Options

yes|true
When set to "yes", login hits are tracked in the registry instead of only in the local process cache.

no|false
When set to "no", login hits are not tracked in the registry instead of only in the local process cache.

Usage

This stanza entry is optional.

Default value

The value is not specified by default during WebSEAL configuration. When the value is not specified, the default value is no.

Example

```
login-failures-persistent = yes
```

max-search-size

Syntax

```
max-search-size = {0|number_entries}
```

Description

Limit for the maximum search size, specified as the number of entries, that can be returned from the LDAP server. The value for each server can be different, depending on how the server was configured.

Options

0
The number is unlimited; there is no limit to the maximum search size.

number_entries
The maximum number of entries for search, specified as an integer whole number. This value can be limited by the LDAP server itself.

Usage

This stanza entry is optional.

Default value

The default value is always taken (during WebSEAL initialization) from the corresponding parameter in the **[ldap]** stanza of the `ldap.conf` configuration file for the LDAP server.

Example

```
max-search-size = 2048
```

prefer-readwrite-server

Syntax

```
prefer-readwrite-server = {yes|true|no|false}
```

Description

Allows or disallows the client to question the Read/Write LDAP server before querying any replica Read-only servers configured in the domain.

Options

yes|true
Enable the choice.

no|false
Disable the choice. Anything other than `yes` | `true`, including a blank value, is interpreted as `no` | `false`.

Usage

This stanza entry is optional.

Default value

no

Example

```
prefer-readwrite-server = no
```

port

Syntax

```
port = port_number
```

Description

Number of the TCP/IP port used for communicating with the LDAP server. Note that this is *not* for SSL communication.

Options

port_number
A valid port number is any positive integer that is allowed by TCP/IP and that is not currently being used by another application.

Usage

This stanza entry is required when LDAP is enabled.

Default value

The default value is always taken (during WebSEAL initialization) from the corresponding parameter in the **[ldap]** stanza of the `ldap.conf` configuration file for the LDAP server.

Example

```
port = 389
```

pwd-chg-method

This options is used to override the automatically used low level LDAP operations for changing account passwords.

Syntax

```
pwd-chg-method = {automatic|mod_pwd_ext_op|del_add|del_then_add|replace}
```

Description

Overrides the automatically used low level LDAP operations for changing account passwords.

Options

automatic
The LDAP server type is detected and the appropriate LDAP password change operations are selected. This is the default behavior.

mod_pwd_ext_op
The **ldap_extended_operation** 1.3.6.1.4.1.4203.1.11.1 is used.

del_add
A single **ldap_modify** is used that has both a **DELETE** for the old password and an **ADD** for the new password.

del_then_add
Two **ldap_modify** invocations are used. The first does a **DELETE** for the old password. The second does an **ADD** for the new password.

replace
A single **ldap_modify** is used, which does a **REPLACE** of the password attribute with the new password.

Usage

This stanza entry is optional.

Default value

`automatic`

Example

```
pwd-chg-method = automatic
```

replica

Syntax

```
replica = ldap-server, port, type, pref
```

Description

Definition of the LDAP user registry replicas in the domain.

Security Verify Access supports a maximum of one host and nine LDAP replica servers, which are listed in the `ldap.conf` file. If more than nine LDAP replica entries are listed, the Security Verify Access servers cannot start.

Options

ldap-server
The network name of the server.

port
The port number for the LDAP server. A valid port number is any positive number that is allowed by TCP/IP and that is not currently being used by another application.

type
One of read-only or read/write.

pref
A number from 1 to 10 (10 is the highest preference).

Usage

This stanza entry is optional.

Default value

Default value is that no replicas are specified.

Any value is always taken during WebSEAL initialization from the corresponding parameter in the **[ldap]** stanza of the `ldap.conf` configuration file for the LDAP server.

Example

Example of one replica specified and two replicas commented out:

```
replica = rep1,390,readonly,1
#replica = rep2,391,readwrite,2
#replica = rep3,392,readwrite,3
```

search-timeout

Syntax

```
search-timeout = {0|number_seconds}
```

Description

Amount of time (in seconds) that will be allowed for search operations before the LDAP server is considered to be down. If specified, this value overrides any value of **timeout** for search operations.

Note: Do not specify this parameter in the `ldap.conf` server configuration file.

Options

0
No timeout is allowed.

number_seconds
The specified number of seconds allowed for search operations, specified as an integer positive whole number. There is no range limitation for timeout values.

Usage

This stanza entry is optional.

Default value

`0`

Example

```
search-timeout = 0
```

ssl-enabled

Syntax

```
ssl-enabled = {yes|true|no|false}
```

Description

Enables or disables SSL communication between WebSEAL and the LDAP server.

Options

yes|true
Enable SSL communication.

no|false
Disable SSL communication.

Usage

This stanza entry is optional.

Default value

SSL communication is disabled by default. During WebSEAL server configuration, the WebSEAL administrator can choose to enable it.

Example

```
ssl-enabled = yes
```

ssl-keyfile

Syntax

```
ssl-keyfile = file_name
```

Description

SSL key file name. The SSL key file handles certificates that are used in LDAP communication.

Options

file_name
The WebSEAL administrator specifies this file name during WebSEAL configuration. The file name can be any arbitrary choice, but the extension is usually `.kdb`.

Usage

This stanza entry is required when SSL communication is enabled, as specified in the **`ssl-enabled`** stanza entry.

Default value

None.

Example

Example:

```
ssl-keyfile = webseald.kdb
```

ssl-keyfile-dn

Syntax

```
ssl-keyfile-dn = key_label
```

Description

String that specifies the key label of the client personal certificate within the SSL key file. This key label is used to identify the client certificate that is presented to the LDAP server.

Options

key_label
String that specifies the key label of the client personal certificate within the SSL key file.

Usage

This stanza entry is optional. A label is not required when one of the certificates in the keyfile has been identified as the default certificate. The decision whether to identify a certificate as the default was made previously by the LDAP administrator when configuring the LDAP server. The WebSEAL configuration utility prompts the WebSEAL administrator to supply a label. When the administrator knows that the certificate contained in the keyfile is the default certificate, the administrator does not have to specify a label.

Default value

None.

Example

```
ssl-keyfile-dn = "PD_LDAP"
```

ssl-port

Syntax

```
ssl-port = port_number
```

Description

SSL IP port that is used to connect to the LDAP server. Note that this is for SSL communication.

Options

port_number
A valid port number is any positive number that is allowed by TCP/IP and that is not currently being used by another application.

Usage

This stanza entry is required only when LDAP is enabled and the LDAP server is configured to perform client authentication (`ssl-enabled = yes`).

Default value

The default value is always taken (during WebSEAL initialization) from the corresponding parameter in the **[ldap]** stanza of the `ldap.conf` configuration file for the LDAP server.

Example

```
ssl-port = 636
```

timeout

Syntax

```
timeout = {0|number_seconds}
```

Description

Amount of time (in seconds) that is allowed for authentication or search operations before the LDAP server is considered to not available. If specified, a value for the stanza entries **authn-timeout** or **search-timeout** overrides the value of this stanza entry.

Note: Do not specify this parameter in the `ldap.conf` server configuration file.

Options

0
No timeout is allowed.

number_seconds
The number of seconds allowed for authentication or search, specified as a positive integer whole number. There is no range limitation for timeout values.

Usage

This stanza entry is optional.

Default value

`0`

Example

```
timeout = 0
```

user-and-group-in-same-suffix

Syntax

```
user-and-group-in-same-suffix = {yes|true|no|false}
```

Description

Indicates whether the groups, in which a user is a member, are defined in the same LDAP suffix as the user definition.

When a user is authenticated, the groups in which the user is a member must be determined in order to build a credential. Normally, all LDAP suffixes are searched to locate the groups of which the user is a member.

Options

yes|true
The groups are assumed to be defined in same LDAP suffix as the user definition. Only that suffix is searched for group membership. This behavior can improve the performance of group lookup because only a single suffix is searched for group membership. This option should only be specified if group definitions are restricted to the same suffix as the user definition.

no|false
The groups might be defined in any LDAP suffix.

Usage

This stanza entry is optional.

Default value

The value is not specified by default during WebSEAL configuration. When the value is not specified, the default value is no.

Example

```
user-and-group-in-same-suffix = yes
```

[local-apps] stanza

application

Use the `application` stanza entry to enable specific embedded applications and associate these applications with a URI path.

Syntax

```
app-name = app-path
```

Description

Entries in this stanza map an embedded application to a URI path segment. The path should include a single path segment (i.e. should not contain a '/' character), which is relative to the root of the local junction (the root of the local junction is usually '/').

Available applications include:

azn-decision
A REST API which can be used to evaluate authorization decisions.

cred-viewer
An API which can be used to return a JSON representation of the local user credential.

jwks
An API which returns the public keys contained in the key database used for junction communication. The response data will conform to RFC 7517.

jct-status
An API which can be used to return the current status, in JSON format, of hosted junctions.

Options

app-name
The name of the application. Possible values include: azn-decision, cred-viewer.

app-path
The path segment, relative to the root of the local junction, which will be used for this application.

Usage

This stanza entry is optional.

Default Value

None.

[local-response-macros] stanza

macro

Syntax

```
macro = macro[:name]
```

Description

URL-encoded macros to include in the query string for all redirected management page requests. WebSEAL provides a default set of macros.

By default, WebSEAL uses the *macro* values as arguments in the generated query string. Alternatively, you can customize the name of the arguments used in the query string by adding a colon followed by a *name* value.

Options

macro
URL-encoded macro.

name
WebSEAL uses this custom name as an argument in the response URI. If you do not provide a value for this custom *name* then WebSEAL defaults to using the *macro* value as an argument in the response URI.

Note: For the HTTPHDR macro, the default value is HTTPHDR_<*name*>, where <*name*> is the name of the HTTP header defined in the macro. For the CREDATTR macro, the default value is CREDATTR_<*name*>, where <*name*> is the name of the attribute defined in the macro.

Usage

This stanza entry is optional.

Default value

None.

Example

The following entry causes WebSEAL to use the default value USERNAME as an argument in the query string.

```
macro = USERNAME
```

The following entry causes WebSEAL to use the custom value myUserName as an argument in the query string.

```
macro = USERNAME:myUserName
```

[local-response-redirect] stanza

local-response-redirect-uri

Use one or more of this entry to define the URIs to which management page requests are redirected.

Syntax

```
local-response-redirect-uri = URI
```

Description

All requests for management pages are redirected to these URIs with a query string that indicates the requested operation, along with any macros (as configured in the **[local-response-macros]** stanza).

You can customize this configuration item for a particular junction by adding the adjusted configuration item to a **[local-response-redirect:*{junction_name}*]** stanza, where *{junction_name}* refers to the junction point for a standard junction (including the leading / character) or the virtual host label for a virtual host junction.

Options

URI

URI to which management page requests are redirected.

The URI can be absolute or server-relative. Use an absolute URI only if the destination server is not accessed through WebSEAL.

Valid formats are as follows:

```
http[s]://<server>/<path>
/<path>
```

To define the URI for specific operations, prefix the URI in the entry with the operation name in the form [*<operation>*]. The "[" and "]" characters are required. Valid values for *<operation>* are as follows:

```
logout
passwd
passwd_warn
passwd_warn_failure
acct_inactivated
acct_locked
passwd_exp
passwd_rep_success
passwd_rep_failure
help
login
login_success
token_login
cert_login
next_token
switch_user
failed_cert
cert_stepup_http
stepup
error
too_many_sessions
tempsession
```

If an entry that does not specify an operation is present, then any operation without a specific entry uses it. If an entry that does not specify an operation is not present, then any operation without a specific entry does not use local response redirection and uses regular WebSEAL behavior instead.

Usage

This stanza entry is optional.

Default value

None.

Example

Example of a server-relative URI:

```
local-response-redirect-uri = /jct/page.html
```

Example of an absolute URI:

```
local-response-redirect-uri = http://www.example.com/
```

Example of an operation-specific URI:

```
local-response-redirect-uri = [login] /jct/cgi-bin/eai
```

[local-response-redirect:<jct-id>] stanza

local-response-redirect-uri

Use one or more of this entry to define the URIs to which management page requests are redirected.

Syntax

```
local-response-redirect-uri = URI
```

Description

All requests for management pages are redirected to these URIs with a query string that indicates the requested operation, along with any macros (as configured in the **[local-response-macros]** stanza).

You can customize this configuration item for a particular junction by adding the adjusted configuration item to a **[local-response-redirect:*{junction_name}*]** stanza, where *{junction_name}* refers to the junction point for a standard junction (including the leading / character) or the virtual host label for a virtual host junction.

Options

URI

URI to which management page requests are redirected.

The URI can be absolute or server-relative. Use an absolute URI only if the destination server is not accessed through WebSEAL.

Valid formats are as follows:

```
http[s]://<server>/<path>
/<path>
```

To define the URI for specific operations, prefix the URI in the entry with the operation name in the form [*<operation>*]. The "[" and "]" characters are required. Valid values for *<operation>* are as follows:

```
logout
passwd
passwd_warn
```

```
passwd_warn_failure
acct_inactivated
acct_locked
passwd_exp
passwd_rep_success
passwd_rep_failure
help
login
login_success
token_login
cert_login
next_token
switch_user
failed_cert
cert_stepup_http
stepup
error
too_many_sessions
tempsession
```

If an entry that does not specify an operation is present, then any operation without a specific entry uses it. If an entry that does not specify an operation is not present, then any operation without a specific entry does not use local response redirection and uses regular WebSEAL behavior instead.

Usage

This stanza entry is optional.

Default value

None.

Example

Example of a server-relative URI:

```
local-response-redirect-uri = /jct/page.html
```

Example of an absolute URI:

```
local-response-redirect-uri = http://www.example.com/
```

Example of an operation-specific URI:

```
local-response-redirect-uri = [login] /jct/cgi-bin/eai
```

[logging] stanza

absolute-uri-in-request-log

Use the **absolute-uri-in-request-log** stanza entry to log the absolute URI in the request log, combined log, and HTTP audit records.

Syntax

```
absolute-uri-in-request-log = {yes|no}
```

Description

Log the absolute URI in the request log, combined log, and HTTP audit records. Adds protocol and host to the path.

Options

yes
Log the absolute URI.

no
Do not log the absolute URI.

Usage

This stanza entry is required.

Default value

no

Example

```
absolute-uri-in-request-log = no
```

agents

Use the **agents** stanza entry to enable or disable the agents log.

Syntax

```
agents = {yes|no}
```

Description

Enables or disables the agents log. This log records the contents of the **User_Agent:** header of each HTTP request.

Options

yes
The value yes enables the agents log.

no
The value no disables the agents log.

Usage

This stanza entry is required.

Default value

yes

Example

```
agents = yes
```

audit-mime-types

Use the **audit-mime-types** stanza entry to configure WebSEAL to use the mime type to determine whether to generate an audit event for a particular HTTP request.

Syntax

```
mime-pattern = {yes|no}
```

Description

WebSEAL determines whether the content-type of the HTTP response matches any of the configured MIME patterns. If the HTTP response does match one of the MIME patterns, WebSEAL uses this entry to determine whether to generate an audit event.

Note:

1. More specific MIME patterns take precedence over less specific MIME patterns. For example, if `image/* = yes` (general), but `image/jpeg = no` (more specific), then an HTTP response with an image MIME-type other than JPEG generates an audit event; a response with a JPEG MIME-type does not generate an audit event.
2. If an HTTP response does not match any of the MIME patterns that are listed in this stanza, WebSEAL does not generate an audit event.

Options

yes
WebSEAL generates an audit event for a response that contains the corresponding content MIME-type.

no
WebSEAL does not generate an audit event for a response that contains the corresponding content MIME-type.

Usage

This stanza entry is optional.

Default value

None.

Example

```
image/jpeg = no
image/* = no
*/* = no
```

audit-response-codes

Use the **audit-response-codes** stanza entry to control whether WebSEAL generates an audit event that is based on the response code of the HTTP response.

Syntax

```
code = {yes|no}
```

Description

Determines whether WebSEAL generates an audit event for an HTTP request that is based on the response code of the HTTP response.

Options

yes
WebSEAL generates an audit event for an HTTP response that matches the corresponding response code.

no
WebSEAL does *not* generate an audit event for an HTTP response that matches the corresponding response code.

Usage

This stanza entry is optional.

Default value

None.

Example

```
200 = no
304 = no
401 = yes
```

flush-time

Use the **flush-time** stanza entry to specify the frequency at which WebSEAL flushes log buffers.

Syntax

```
flush-time = number_of_seconds
```

Description

Integer value indicating the frequency, in seconds, to force a flush of log buffers.

Options

number_of_seconds
Integer value indicating the frequency, in seconds, to force a flush of log buffers. The minimum value is 1 second. The maximum value is 600 seconds.

Usage

This stanza entry is optional.

Default value

20

Example

```
flush-time = 20
```

gmt-time

Syntax

```
gmt-time = {yes|no}
```

Description

Enables or disables logging requests using Greenwich Mean Time (GMT) instead of the local timezone.

Options

yes
A value of yes means to use GMT

no
A value of no means to use the local timezone.

Usage

This stanza entry is required.

Default value

no

Example

```
gmt-time = no
```

host-header-in-request-log

Syntax

```
host-header-in-request-log = {yes|no}
```

Description

Log the **Host** header at the front of each line in the request log and the combined log.

Options

yes
Log the **Host** header.

no
Do not log the **Host** header.

Usage

This stanza entry is required.

Default value

no

Example

```
host-header-in-request-log = no
```

log-invalid-requests

Syntax

```
log-invalid-requests = {yes|no}
```

Description

Specifies whether or not WebSEAL logs all requests that are malformed or for some other reason is not processed to completion.

Options

yes
WebSEAL logs every request, even if a request is malformed or for some other reason is not processed to completion.

no
WebSEAL logs most requests. In some cases, requests that are malformed or for some other reason are not processed to completion will not be logged. This option exists for compatibility with versions of WebSEAL prior to version 6.0.

Usage

This stanza entry is required.

Default value

yes

Example

```
log-invalid-requests = yes
```

max-size

Syntax

```
max-size = number_of_bytes
```

Description

Integer value indicating the size limit of the log files. This value applies to the request, referer, and agent logs. The size limit is also referred to as the *rollover threshold*. When the log file reaches this threshold, the original log file is renamed and a new log file with the original name is created.

Options

number_of_bytes

When the value is zero (0), no rollover log file is created.

When the value is a negative integer, the logs are rolled over daily, regardless of the size.

When the value is a positive integer, the value indicates the maximum size, in bytes, of the log file before the rollover occurs. The allowable range is from 1 byte to 2 gigabytes.

Usage

This stanza entry is required.

Default value

2000000

Example

```
max-size = 2000000
```

referers

Syntax

```
referers = {yes|no}
```

Description

Enables or disables the referers log. This log records the **Referer:** header of each HTTP request.

Options

yes
The value yes enables referers logging.

no
The value no disables referers logging.

Usage

This stanza entry is required.

Default value

yes

Example

```
referers = yes
```

requests

Syntax

```
requests = {yes|no}
```

Description

Enables or disables the requests log. This log records standard logging of HTTP requests.

Options

yes
The value yes enables requests logging.

no
The value no disables requests logging.

Usage

This stanza entry is required.

Default value

yes

Example

```
requests = yes
```

request-log-format

Syntax

```
request-log-format = directives
```

Description

Contains the format in which a customized request log should be created.

Options

There are two form of directives:

%directive
The value of the directive will be written to the request log.

%!directive
SHA256 hash of the value will be written to the request log.

The following directives can be used:

%a
Remote IP Address.

%A
Local IP Address.

%b
Bytes in the reply excluding HTTP headers in CLF format: '-' instead of 0 when no bytes are returned.

%B
Bytes in the reply excluding HTTP headers.

%{Attribute}C
Attribute from the Security Verify Access credential named "*Attribute*".

%c
The HTTP response status received from the junctioned server.

%{cookie-name}e
Contents of the cookie "*cookiename*" in the request.

%{cookie-name}E
Contents of the cookie "*cookiename*" in the response.

%d
Transaction identifier, or session sequence number.

%F
Time taken to serve the request in microseconds.

%h
Remote host.

%H
Request protocol.

%{header-name}i
Contents of the Header *header-name* in the request.

%j
The name of the junction in the request.

%J
The length of time, in microseconds, that the junction server spent processing the request. This will include the time that it took to send the request to the server, the length of time that it took the server to process the request, and the length of time that it took to read and process the response header.

%l
Remote logname.

%m
Request method (that is, GET, POST, HEAD).

%M
The time, in Common Log Format, at which the request was received with millisecond precision.

%{header-name}o
Contents of the Header *header-name* in the reply.

%p
Port of the WebSEAL server the request was served on.

%q
The query string (prepended with '?' or empty).

%Q
Logs raw query strings that the user must decode manually.

%r
First line of the request.

%R
First line of the request including HTTP://HOSTNAME.

%s
Status.

%S
The hostname of the junctioned server which serviced this request.

%t
Time and date in CLF format.

%{format}t
The time and date in the given format.

The "format" is the same format options as the UNIX "date" command. For example:

%a
Displays the locale's abbreviated weekday name.

%A
Displays the locale's full weekday name.

%b
Displays the locale's abbreviated month name.

%B
Displays the locale's full month name.

%c
Displays the locale's appropriate date and time representation. This is the default.

%C
Displays the first two digits of the four-digit year as a decimal number (00-99). A year is divided by 100 and truncated to an integer.

%d
Displays the day of the month as a decimal number (01-31). In a two-digit field, a 0 is used as leading space fill.

%D
Displays the date in the format equivalent to %m/%d/%y.

%e
Displays the day of the month as a decimal number (1-31). In a two-digit field, a blank space is used as leading space fill.

%h
Displays the locale's abbreviated month name (a synonym for %b).

%H
Displays the hour (24-hour clock) as a decimal number (00-23).

%I
Displays the hour (12-hour clock) as a decimal number (01-12).

%j
Displays the day of year as a decimal number (001-366).

%k
Displays the 24-hour-clock hour clock as a right-justified, space-filled number (0 to 23).

%m
Displays the month of year as a decimal number (01-12).

%M
Displays the minutes as a decimal number (00-59).

%n
Inserts a <new-line> character.

%p
Displays the locale's equivalent of either AM or PM.

%r
Displays 12-hour clock time (01-12) using the AM-PM notation; in the POSIX locale, this is equivalent to %I:%M:%S %p.

%S
Displays the seconds as a decimal number (00- 59).

%s
Displays the number of seconds since January 1, 1970, Coordinated Universal Time (CUT).

%t
Inserts a <tab> character.

%T
Displays the 24-hour clock (00-23) in the format equivalent to HH:MM:SS.

%u
Displays the weekday as a decimal number from 1-7 (Sunday = 7). Refer to the %w field descriptor.

%U
Displays week of the year(Sunday as the first day of the week) as a decimal number[00 - 53] . All days in a new year preceding the first Sunday are considered to be in week 0.

%V
Displays the week of the year as a decimal number from 01-53 (Monday is used as the first day of the week). If the week containing January 1 has four or more days in the new year, then it is considered week 01; otherwise, it is week 53 of the previous year.

%w
Displays the weekday as a decimal number from 0-6 (Sunday = 0). Refer to the %u field descriptor.

%W
Displays the week number of the year as a decimal number (00-53) counting Monday as the first day of the week.

%x
Displays the locale's appropriate date representation.

%X
Displays the locale's appropriate time representation.

%y
Displays the last two numbers of the year (00-99).

%Y
Displays the four-digit year as a decimal number.

%Z
Displays the time-zone name, or no characters if no time zone is determinable.

%%
Displays a % (percent sign) character.

%T
Time taken to serve the request in seconds.

%u
Remote user.

%U
The URL requested.

%v
Canonical ServerName of the server serving the request.

%{env-name}V
Contents of the environment variable *env-name*.

%z
The path portion of the URL in decoded form.

%Z
The path portion of the URL in raw form.

Usage

The `request-log-format` string CANNOT contain the # character.

Default value

The default of this parameter is equivalent to the normal default log output. It is commented out by default.

Example

Example on UNIX or Linux®:

```
request-log-format = %h %l %u %t "%r" %s %b
```

server-log-cfg

Syntax

```
server-log-cfg = agent [parameter=value],[parameter=value]...
```

Description

Configures the server for logging. You can use the available parameters to configure the logging agents.

Options

agent

Specifies the logging agent. The agent controls the logging destination for server events. Valid agents include:

- `stdout`
- `stderr`
- `file`
- `remote`
- `rsyslog`

Note: If you use the `remote` agent to send audit events to a remote authorization server, ensure that the destination server is configured to process the received events. In particular, the **`logcfg`** configuration entry in the **`aznapi-configuration`** stanza must be set on the remote authorization server. You must use the following format for the `category` value in this **`logcfg`** entry:

```
remote.webseal.hostname.webseald
```

where

hostname

The name of the appliance that originated the event.

For example, the following entry configures the remote authorization server to accept logging events from the `iswga.au.ibm.com` server, and send these events to the `event.log` file:

```
logcfg = remote.webseal.iswga.au.ibm.com.webseald:file path=/var/
PolicyDirector/log/event.log
```

The remote authorization server discards any events that originate from a server for which there is no matching **`logcfg`** rule.

parameter

The different agents support the following configuration parameters:

Table 2. Logging agent configuration parameters

Parameter	Supporting agents
`buffer_size`	`remote`
`compress`	`remote`
`dn`	`remote`

Table 2. Logging agent configuration parameters (continued)	
Parameter	**Supporting agents**
error_retry	remote, rsyslog
flush_interval	all
hi_water	all
log_id	file, rsyslog
max_event_len	rsyslog
max_rollover_files	file
mode	file
path	all
port	remote, rsyslog
queue_size	all
rebind_retry	remote, rsyslog
rollover_size	file
server	remote, rsyslog
ssl_keyfile	rsyslog
ssl_label	rsyslog
ssl_stashfile	rsyslog

Note: For a complete description of the available logging agents and the supported configuration parameters, see the *Security Verify Access: Auditing Guide*.

Usage

This stanza entry is required.

Default value

None.

Example

To log server events in a file called `msg__webseald.log`:

```
server-log-cfg = file path=msg__webseald.log
```

To send server events to a remote syslog server:

```
server-log-cfg = rsyslog server=timelord,port=514,log_id=webseal-instance
```

[ltpa] stanza

Accept and generate LTPA cookies for authentication.

ltpa-auth

Syntax

```
ltpa-auth = {http|https|both|none}
```

Description

Enables support for LTPA cookie generation and authentication.

Options

http
Enables support for http cookies.

https
Enables support for https cookies.

both
Enables support for both http and https cookies.

none
Disables support for both http and https cookies.

Usage

This stanza entry is required.

Default value

none

Example

```
ltpa-auth = https
```

cookie-name

Use the **cookie-name** stanza entry to specify the name of the LTPA cookie that WebSEAL issues to clients.

Syntax

```
cookie-name = cookie_name
```

Description

The name of the LTPA cookie that WebSEAL issues to clients.

Options

cookie_name
This must be Ltpatoken2 as only LTPA version 2 cookies are supported.

Usage

This stanza entry is required.

Default value

Ltpatoken2

Example

```
cookie-name = Ltpatoken2
```

cookie-domain

Use the **cookie-domain** stanza entry to specify the domain of the LTPA cookie that WebSEAL issues to clients.

Syntax

```
cookie-domain = domain_name
```

Description

The domain of the LTPA cookie that WebSEAL issues to clients. If you do not specify a cookie domain, WebSEAL creates the LTPA cookie as a host-only cookie.

Options

domain_name
The domain of the LTPA cookie.

Usage

This stanza entry is required.

Default value

none

Example

```
cookie-domain = ibm.com
```

jct-ltpa-cookie-name

Syntax

```
jct-ltpa-cookie-name = cookie_name
```

Description

The name of the cookie containing the LTPA token that WebSEAL sends across the junction to the backend server. If you do not specify a value for this item, WebSEAL uses the following default values:

- **LtpaToken** for cookies containing LTPA tokens.
- **LtpaToken2** for cookies containing LTPA version 2 tokens.

WebSphere also uses these default values.

You can customize this configuration item for a particular junction by adding the adjusted configuration item to a **[ltpa:<jct_id>]** stanza.

Options

cookie_name
This name must match the LTPA cookie name that the WebSphere application uses on this junction.

Usage

This stanza entry is optional.

Default value

The default value for LTPA tokens is `LtpaToken`.

The default value for LTPA2 tokens is `LtpaToken2`.

Example

```
jct-ltpa-cookie-name = myCookieName
```

keyfile

Syntax

```
keyfile = keyfile_name
```

Description

The key file used when accessing LTPA cookies. The value must correspond to a valid LTPA key file, as generated by WebSphere.

Options

keyfile_name
Name of a valid LTPA key file, as generated by WebSphere.

Usage

This stanza entry is optional.

Default value

none

Example

```
keyfile = keyfile123
```

update-cookie

Syntax

```
update-cookie = number_of_seconds
```

Description

The number of seconds that pass between updates of the LTPA cookie with the lifetime of the cookie.With each request, if *n* seconds have passed since the last cookie update, another update will occur. A zero value will cause the lifetime timestamp in the LTPA cookie to be updated with each request.Negative values will cause the lifetime of the cookie to be set to the same value as the lifetime of the user session.This setting is used in an attempt to mimic the inactivity timeout of a user session.

Note: This configuration entry affects the LTPA cookie that WebSEAL issues to clients. It is the lifetime of the cookie specified by the **cookie-name** configuration entry in the **[ltpa]** stanza.

Options

number_of_seconds
The number of seconds that pass between updates of the LTPA cookie with the lifetime of the cookie.

Usage

This stanza entry is required.

Default value

`-1`

Example

```
update-cookie = 0
```

use-full-dn

Syntax

```
use-full-dn = {true|false}
```

Description

Controls whether the generated LTPA cookie contains the full DN of the user, or the Security Verify Access short name of the user.

Options

`true`
WebSEAL inserts the full DN of the user into the LTPA cookie.

`false`
WebSEAL inserts the Security Verify Access short name of the user into the LTPA cookie.

Usage

This stanza entry is optional.

Default value

`true`

Example

```
use-full-dn = true
```

[ltpa:<jct-id>] stanza

jct-ltpa-cookie-name

Syntax

```
jct-ltpa-cookie-name = cookie_name
```

Description

The name of the cookie containing the LTPA token that WebSEAL sends across the junction to the backend server. If you do not specify a value for this item, WebSEAL uses the following default values:

- **LtpaToken** for cookies containing LTPA tokens.
- **LtpaToken2** for cookies containing LTPA version 2 tokens.

WebSphere also uses these default values.

You can customize this configuration item for a particular junction by adding the adjusted configuration item to a **[ltpa:<jct_id>]** stanza.

Options

cookie_name
This name must match the LTPA cookie name that the WebSphere application uses on this junction.

Usage

This stanza entry is optional.

Default value

The default value for LTPA tokens is `LtpaToken`.

The default value for LTPA2 tokens is `LtpaToken2`.

Example

```
jct-ltpa-cookie-name = myCookieName
```

[ltpa-cache] stanza

ltpa-cache-enabled

Syntax

```
ltpa-cache-enabled = {yes|no}
```

Description

Enables or disables the Lightweight Third Party Authentication cache.

Options

yes
A value of yes enables caching.

no
A value of no disables caching.

Usage

This stanza entry is required.

Default value

yes

Example

```
ltpa-cache-enabled = yes
```

ltpa-cache-entry-idle-timeout

Syntax

```
ltpa-cache-entry-idle-timeout = number_of_seconds
```

Description

Integer value that specifies the timeout, in seconds, for cache entries that are idle.

Options

number_of_seconds
Integer value that specifies the timeout, in seconds, for cache entries that are idle. The value must be greater than or equal to zero (0). A value of zero means that entries are not removed from the LTPA cache due to inactivity. However, they may still be removed due to either the **ltpa-cache-size** being exceeded or the **ltpa-cache-entry-lifetime** stanza entry being exceeded. WebSEAL does not impose a maximum value.

Usage

This stanza entry is required, but is ignored when LTPA caching is disabled.

Default value

600

Example

```
ltpa-cache-entry-idle-timeout = 600
```

ltpa-cache-entry-lifetime

Syntax

```
ltpa-cache-entry-lifetime = number_of_seconds
```

Description

Integer value that specifies the lifetime, in seconds, of a LTPA cache entry.

Options

number_of_seconds

Integer value that specifies the lifetime, in seconds, of a LTPA cache entry. The value must be greater than or equal to zero (0). A value of zero means that entries are not removed from the LTPA cache due to their entry lifetime being exceeded. However, they may still be removed due to either the **ltpa-cache-size** being exceeded or the **ltpa-cache-entry-idle-timeout** stanza entry being exceeded. WebSEAL does not impose a maximum value.

Usage

This stanza entry is required, but is ignored when LTPA caching is disabled.

Default value

3600

Example

```
ltpa-cache-entry-lifetime = 3600
```

ltpa-cache-size

Syntax

```
ltpa-cache-size = number_of_entries
```

Description

Integer value indicating the number of entries allowed in the LTPA cache.

Options

number_of_entries

Integer value indicating the number of entries allowed in the LTPA cache. The value must be greater than or equal to zero (0). A value of zero means that there is no limit on the size of the LTPA cache. This is not recommended.

WebSEAL does not impose a maximum value. Choose your maximum value to stay safely within the bounds of your available system memory.

Usage

This stanza entry is required, but is ignored when LTPA caching is disabled.

Default value

4096

Example

```
ltpa-cache-size = 4096
```

[mpa] stanza

mpa

Syntax

```
mpa = {yes|no}
```

Description

Enables support for multiplexing proxy agents.

Options

yes
Enables support for multiplexing proxy agents.

no
Disables support for multiplexing proxy agents.

Usage

This stanza entry is required.

Default value

no

Example

```
mpa = no
```

[oauth] stanza

Use the **[oauth]** stanza to configure Open Authentication (OAuth) settings.

cluster-name

Use the **cluster-name** entry in the **[oauth]** stanza to specify the Federation Runtime cluster that hosts the OAuth service.

Syntax

```
cluster-name = <cluster>
```

Description

The name of the Federation Runtime cluster that hosts this OAuth service. You must also specify a corresponding **[tfim-cluster:<*cluster*>]stanza**, which contains the definition of the cluster.

Options

<cluster>
The name of the Federation Runtime cluster where the OAuth service is hosted.

Usage

This stanza entry is required when you configure OAuth authentication.

Default value

None.

Example

```
cluster-name = oauth-cluster
```

For this example, there needs to be a corresponding **[tfim-cluster:oauth-cluster]** stanza to define the cluster.

continue-on-auth-failure

Use the **continue-on-auth-failure** stanza entry to define whether to continue processing the request when authentication fails.

Syntax

```
continue-on-auth-failure = {yes|true|no|false}
```

Description

This stanza entry controls whether to continue processing the request and try additional authentication mechanisms if an invalid authorization header has been supplied with the request.

Options

yes|true
Continue processing the request and try additional authentication mechanisms.

no|false
Do not continue processing the request.

Usage

This stanza entry is optional.

Default value

`false`

Example

```
continue-on-auth-failure = false
```

external-group-attribute

Use the **external-group-attribute** entry to indicate which STSUU attribute in the RSTR contains the external group identities that are used when authenticating an external user. Remove this configuration entry if you do not want to allow authentication using an external group identity.

Syntax

```
external-group-attribute = attribute_name
```

Description

If this entry is set, the appliance searches for an external group identity in the STSUU. If an external group identity is present, it will be used without further changes. If this entry is not configured, an external group identity cannot be used to authenticate the user. You can specify multiple attributes in the form of a comma separated list.

The group information is only used if the user is authenticating as an external user.

Options

attribute_name
The name of the external group identity attribute to be extracted from the RSTR. .

Usage

This stanza entry is optional.

Default value

`am-ext-user-groups`

Example

```
external-group-attribute = am-ext-user-groups
```

external-user-identity-attribute

Use the **external-user-identity-attribute** entry to indicate which STSUU attribute in the RSTR contains the external user identity that can perform OAuth authentication. Remove this configuration entry if you do not want to allow authentication using an external user identity.

Syntax

```
external-user-identity-attribute = attribute_name
```

Description

If this entry is set, the appliance searches for an external user identity in the STSUU. If an external user identity is present, it will be used without further changes. If this entry is not configured, an external user identity cannot be used to authenticate the user.

Options

attribute_name
The name of the external user identity attribute to be extracted from the RSTR.

Usage

This stanza entry is optional.

Default value

`am-ext-user-id`

Example

```
external-user-identity-attribute = am-ext-user-id
```

default-fed-id

Use the **default-fed-id** stanza entry to specify the default federation provider that WebSEAL uses for OAuth requests.

Syntax

```
default-fed-id = <provider_url>
```

Description

The Provider ID of the default OAuth federation in Federation Runtime. By default, WebSEAL uses this provider ID for OAuth requests.

You can override this default provider for an individual request by including a request parameter that has the name specified by the **fed-id-param** configuration entry.

Options

<provider_url>
The IP address for the federation provider that WebSEAL uses for OAuth requests. You can find the Provider ID of a federation on the federation properties page.

Usage

This stanza entry is required when you configure OAuth authentication.

Default value

None

Example

```
default-fed-id = https://localhost/sps/oauthfed/oauth10
```

fed-id-param

Use the **fed-id-param** stanza entry to specify the name of the request parameter whose value specifies the Provider ID for WebSEAL to include in OAuth requests.

Syntax

```
fed-id-param = <request_param_name>
```

Description

The name of the parameter that you can include in a request to override the Provider ID that is specified by the **default-fed-id** configuration entry. If this **fed-id-param** configuration entry is set, WebSEAL checks incoming requests for a parameter with the specified name. If this request parameter exists, WebSEAL uses the Provider ID contained in the request rather than the **default-fed-id** Provider ID.

Note: You can delete this configuration entry to ensure that WebSEAL always uses the default provider that is specified by **default-fed-id**.

Options

<request_param_name>
The name of the request parameter whose value specifies the Provider ID for WebSEAL to include in OAuth requests. If no such parameter exists in the request, WebSEAL uses the Provider ID specified by **default-fed-id**.

Usage

This stanza entry is optional.

Note: If you do not configure this stanza entry, WebSEAL always uses the provider that is configured as the **default-fed-id**.

Default value

None.

Example

```
fed-id-param = FederationId
```

multivalue-scope

Use the **multivalue-scope** stanza entry to specify whether OAuth multivalue scope support is enabled.

Syntax

```
multivalue-scope = {true | false}
```

Description

The **multivalue-scope** entry enables or disables OAuth multivalue scope.

Options

true
OAuth multivalue scope is enabled.

false
OAuth multivalue scope is disabled.

Usage

This stanza entry is optional.

Default value

The default value is false. By default, OAuth multivalue scope is disabled.

Example

```
multivalue-scope = false
```

oauth-auth

Use the **oauth-auth** stanza entry to specify whether Open Authentication (OAuth) is enabled.

Syntax

```
oauth-auth = {http | https | both | none}
```

Description

Enables authentication with OAuth mechanism.

Options

http
Enable OAuth over HTTP.

https
Enable OAuth over HTTPS.

both
Enable OAuth over both HTTP and HTTPS.

none
Disable OAuth.

Usage

This stanza entry is required when you configure OAuth authentication.

Default value

The default value is none. By default, OAuth authentication is disabled.

Example

```
oauth-auth = none
```

pac-attribute

Use the **pac-attribute** entry to indicate which STSUU attribute in the RSTR contains a PAC to be used for authentication. Remove this configuration entry if you do not want to allow authentication using a PAC.

Syntax

```
pac-attribute = attribute_name
```

Description

If this entry is set, the appliance searches for a PAC in the STSUU. The PAC takes precedence over all other authentication data. If a PAC is present, it will be used without further changes. If this entry is not configured, a PAC cannot be used to authenticate the user.

Options

attribute_name
The name of the PAC attribute to be extracted from the RSTR.

Usage

This stanza entry is optional.

Default value

`am-pac`

Example

```
pac-attribute = am-pac
```

user-identity-attribute

Use the **`user-identity-attribute`** stanza entry to specify the attribute that contains the user identity.

Syntax

```
user-identity-attribute = <attribute_name>
```

Description

Specifies which attribute, within the RSTR from IBM Security Verify Access, contains the user identity. The value of this entry is used as the principal when a credential is created.

Options

<attribute_name>
The attribute that contains the user identity.

Usage

This stanza entry is required.

Default value

The default value is `username`.

Example

```
user-identity-attribute = username
```

[oauth-eas] stanza

Notes:

- You can configure this stanza to support OAuth authorization decisions as part of WebSEAL requests. For more information about OAuth authorization decisions support, see the *IBM Security Verify Access: Web Reverse Proxy Configuration Guide*.
- The OAuth EAS is used for a particular object if the effective POP for the object has an attribute called **`eas-trigger`**, with an associated value of `trigger_oauth_eas`.

allow-query-string-token

Use the `allow-query-string-token` stanza entry to control whether the authentication token can be obtained from the query string.

Syntax

```
allow-query-string-token = {true | false}
```

Description

The original OAuth specification allowed the authentication token to be obtained from the query string. The specification has since been adjusted to remove this option as it can be a security risk to embed authentication information within the URL. This configuration entry can be used to control whether authentication tokens found in the query string will be used.

Options

`true`
The OAuth EAS will search the query string for authentication tokens.

`false`
The OAuth EAS will ignore any authentication tokens found in the query string.

Usage

Optional.

Default value

`false`

Example

```
allow-query-string-token = true
```

apply-tam-native-policy

Use the **`apply-tam-native-policy`** stanza entry to control whether the OAuth EAS also uses the native Security Verify Access policy when it is determining the access permissions for the resource.

Syntax

```
apply-tam-native-policy = {true | false}
```

Description

Determines whether the Security Verify Access policy still takes effect, in addition to the OAuth authorization.

Options

true
The OAuth EAS checks the Security Verify Access policy to determine whether the user has permission to access the resource.

false
The OAuth EAS does not check the Security Verify Access ACL policy to determine whether the user has permission to access the resource.

Usage

This stanza entry is required when you configure OAuth EAS authentication.

Default value

None.

Example

```
apply-tam-native-policy = false
```

bad-gateway-rsp-file

Use the **bad-gateway-rsp-file** stanza entry to specify the file that contains the content that WebSEAL uses to construct a `502 Bad Gateway` response.

Syntax

```
bad-gateway-rsp-file = <file_name>
```

Description

Specifies the file that contains the body that WebSEAL uses to construct a `502 Bad Gateway` response. This response is generated when the Federation Runtime fails to process the request.

Options

<file_name>
The name of the 502 Bad Gateway response file.

Usage

This stanza entry is required when you configure OAuth EAS authentication.

Default value

None.

Example

```
bad-gateway-rsp-file = bad_gateway.html
```

bad-request-rsp-file

Use the **bad-request-rsp-file** stanza entry to specify the file that contains the content that WebSEAL uses to construct a `400 Bad Request` response.

Syntax

```
bad-request-rsp-file = <file_name>
```

Description

Specifies the file that contains the body that WebSEAL uses to construct a `400 Bad Request` response. This response is generated when required OAuth elements are missing from a request.

Options

<file_name>
The name of the 400 Bad Request response file.

Usage

This stanza entry is required when you configure OAuth-EAS authentication.

Default value

None.

Example

```
bad-request-rsp-file = bad_rqst.html
```

cache-size

Syntax

```
cache-size = <number_decisions>
```

Description

Specifies the maximum number of OAuth 2.0 bearer token authorization decisions to cache. This EAS has a built-in cache for storing authorization decisions so that WebSEAL can repeatedly use the same OAuth 2.0 bearer token without sending repeated requests to the Federation Runtime.

WebSEAL can cache bearer token decisions because they do not require signing of the request, unlike OAuth 1.0 requests. The lifetime of the cache entry depends on the **Expires** attribute that the Federation Runtime returns. If the Federation Runtime does not return this attribute, WebSEAL does not cache the decision.

This EAS implements a Least Recently Used cache. The decision associated with the least recently used bearer token is forgotten when a new bearer token decision is cached. A cache-size of `0` disables caching of authorization decisions.

Options

<number_decisions>
The maximum number of OAuth 2.0 bearer token authorization decisions that WebSEAL caches.

Usage

This stanza entry is optional.

Default value

The default value is `0`, which disables caching of authorization decisions.

Example

```
cache-size = 0
```

credential-attributes

Use the **credential-attributes** stanza entry to specify any additional attributes from the user credential that should be added to the request that is sent to the server.

Syntax

```
credential-attributes= <credential_attribute>
```

Description

Specifies any additional attributes from the user credential that should be added to the request that is sent to the server. If the specified attribute is missing from the user credential, an attribute with an empty value is added to the request. Multiple attributes can be specified, delimited by a comma (',').

Options

<credential_attribute>
Any additional attributes from the user credential that should be added to the request that is sent to the server.

Usage

This stanza is optional.

Default Value

None.

Example

```
credential-attributes= AZN_CRED_PRINCIPAL_NAME,AZN_CRED_GROUPS
```

default-mode

Use the **default-mode** stanza entry to specify the default OAuth EAS mode.

Syntax

```
default-mode = <oauth_mode>
```

Description

The mode affects the validation of request parameters and the construction of the RequestSecurityToken (RST) sent to the Federation Runtime.

You can override this default mode for an individual request by providing a valid mode value `[OAuth10|OAuth20Bearer]` in a request parameter. The request parameter must have the name that is specified by the **mode-param** configuration entry.

Options

<oauth_mode>
The OAuth mode that the OAuth EAS uses by default.

Usage

This stanza entry is required when you configure OAuth EAS authentication.

Default value

None.

Example

```
default-mode = OAuth10
```

eas-enabled

Use the **eas-enabled** stanza entry to specify whether OAuth external authorization service (EAS) is enabled.

Syntax

```
eas-enabled = {true | false}
```

Description

Enable or disable OAuth EAS.

Options

true
OAuth EAS is enabled.

false
OAuth EAS is disabled.

Usage

This stanza entry is optional.

Default value

The default value is `false`. By default, OAuth EAS is disabled.

Example

```
eas-enabled = true
```

mode-param

The name of the parameter that you can include in a request to override the mode that is specified by the **default-mode** configuration entry.

Syntax

```
mode-param = <mode_name>
```

Description

If this **mode-param** configuration entry is set, WebSEAL checks incoming requests for a parameter with the specified name. If this request parameter exists, WebSEAL uses the mode contained in the request rather than the mode specified by **default-mode**.

Note: You can delete this configuration entry to ensure that WebSEAL always uses the default mode that is specified by **default-mode**.

Options

<mode_name>
The name of the request parameter whose value specifies the mode for OAuth EAS to use. If no such parameter exists in the request, WebSEAL uses the mode specified by **default-mode**.

Usage

This stanza entry is optional.

Note: If you do not configure this stanza entry, WebSEAL always uses the mode that is configured as the **default-mode**.

Default value

None.

Example

```
mode-param = mode
```

realm-name

Syntax

```
realm-name = <realm_name>
```

Description

The name of the OAuth realm that is used in a 401 request for OAuth data.

Options

<realm_name>
The name of the OAuth realm.

Usage

This stanza entry is required when you configure OAuth EAS authentication.

Default value

None.

Example

```
realm-name = realmOne
```

trace-component

Syntax

```
trace-component = <component_name>
```

Description

The name of the Security Verify Access trace component that the OAuth EAS uses.

Options

<component_name>
The name of the Security Verify Access trace component.

Usage

This stanza entry is required when configuring OAuth EAS authentication.

Note: The **pdweb.oauth** component traces the data that passes into the OAuth EAS, which is governed by the **[azn-decision-info]** stanza. This trace might contain sensitive information.

Default value

None.

Example

```
trace-component = pdweb.oauth
```

unauthorized-rsp-file

Syntax

```
unauthorized-rsp-file = <file_name>
```

Description

Specifies the file that contains the body that is used when constructing a `401 Unauthorized` response. This response is generated when either of the following scenarios occur:

- All OAuth data is missing from a request.
- The OAuth data fails validation.

Options

<file_name>
The name of the 401 Unauthorized response file.

Usage

This stanza entry is required when you configure OAuth EAS authentication.

Default value

None.

Example

```
unauthorized-rsp-file = unauth_response.html
```

[oauth-introspection] stanza

The OAuth Introspection capability is configured by using the `[oauth-introspection]` stanza.

The following stanza entries are used to configure the ability for WebSEAL to authenticate an OAuth token using an OAuth introspection endpoint.

The OAuth introspection configuration can be customised for individual junctions by adding configuration entries to a stanza name which is qualified with the junction identifier (i.e. [oauth-introspection:{jct-id}]). The junction identifier refers to the junction point for a standard junction (including the leading '/'), or the virtual host label for a virtual host junction.

auth-method

The introspection request can be authenticated using Basic Authentication or Forms.

Syntax

```
auth-method = {client_secret_post | client_secret_basic}
```

Description

Controls the method by which the authentication information is supplied to the introspection endpoint. This configuration item may be customized for a particular junction by adding the adjusted configuration item to a `[oauth-introspection:`*`{jct_id}`*`]` stanza, where *`'{jct-id}'`* refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

Options

`client_secret_post`
The authentication information is supplied in a form POST.

`client_secret_basic`
The authentication information is provided in an authorization header.

Usage

This stanza entry is required.

Default value

`client_secret_post`

Example

```
auth-method = client_secret_basic
```

client-id

Use the client identifier to specify the client which will be used when authenticating to the introspection endpoint.

Syntax

```
client-id = string
```

Description

The client identifier that is used to authenticate to the introspection endpoint. This configuration item may be customized for a particular junction by adding the adjusted configuration item to a `[oauth-introspection:{jct_id}]` stanza, where `'{jct-id}'` refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

Options

string
The client identifier that is used to authenticate to the introspection endpoint.

Usage

This stanza is optional.

Default value

None.

Example

```
client-id = 56879ef20e75817b329576
```

client-id-hdr

Use the client identifier header to specify the name of the HTTP header to be used when authenticating to the introspection endpoint.

Syntax

```
client-id-hdr = header-name
```

Description

The name of the HTTP header that contains the client identifier that is used to authenticate to the introspection endpoint. This configuration item may be customized for a particular junction by adding the adjusted configuration item to a `[oauth-introspection:{jct_id}]` stanza, where `'{jct-id}'` refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

Options

header-name
The name of the header that contains the client identifier.

Usage

This configuration entry is mutually exclusive with the `client-id` configuration entry. If the `client-id` configuration entry is provided, this configuration entry is ignored.

Default value

None.

Example

```
client-id-hdr = x-IBM-CLIENT-ID
```

client-secret

Use the client-secret to authenticate to the introspection endpoint.

Syntax

```
client-secret = string
```

Description

The client secret that is used to authenticate to the introspection endpoint. If a client-id field is not configured the secret is treated as a bearer token, otherwise it is used in a basic authentication header. This configuration item may be customized for a particular junction by adding the adjusted configuration item to a `[oauth-introspection:{jct_id}]` stanza, where `'{jct-id}'` refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

Options

string
The client secret that is used to authenticate to the introspection endpoint.

Usage

This stanza entry is required.

Default value

None.

Example

```
client-secret = as5dgi92ytapsejguas
```

continue-on-auth-failure

Use the **continue-on-auth-failure** stanza entry to define whether to continue processing the request if authentication fails.

Syntax

```
continue-on-auth-failure = {yes|true|no|false}
```

Description

This stanza entry controls whether to continue processing the request and try additional authentication mechanisms if the introspection has failed. This configuration item may be customized for a particular junction by adding the adjusted configuration item to a `[oauth-introspection:{jct_id}]` stanza, where `'{jct-id}'` refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

Options

yes|true
Continue processing the request and try additional authentication mechanisms.

no|false
Do not continue processing the request on an introspection failure.

Usage

This stanza entry is optional.

Default value

`true`

Example

```
continue-on-auth-failure = false
```

external-user

Use this entry to set whether the mapped identity should correspond to a known Security Verify Access identity.

Syntax

```
external-user = {true | false}
```

Description

This boolean is used to indicate whether the mapped identity should correspond to a known Security Verify Access identity. When you are creating a session with an external user, the credential does not contain any group membership. The scopes contained in the credential should instead be used for authorization policy. This configuration item may be customized for a particular junction by adding the adjusted configuration item to a `[oauth-introspection:{jct_id}]` stanza, where `'{jct-id}'` refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

Options

true
The mapped identity should correspond to a known Security Verify Access identity.

false
The mapped identity does not need to correspond to a known Security Verify Access identity.

Usage

This stanza entry is required.

Example

```
external-user = true
```

http-header

Use the `http-header` stanza entry to add HTTP headers to the OAuth introspection request.

Syntax

```
http-header = <header-name>:<header-data>
```

Description

Controls the addition of HTTP headers into the OAuth introspection request.

This configuration item may be customized for a particular junction by adding the adjusted configuration item to a [oauth-introspection:*{jct_id}*] stanza, where '*{jct-id}*' refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

Multiple headers can be specified by including this configuration entry multiple times.

Options

<header-name>
The name of the HTTP header that holds the data. Valid strings are limited to the following characters: A-Z, a-z, 0–9, hyphen (-), or underscore (_).

<header-data>
The type of data that WebSEAL adds to the <`header-name`> header of the request. The valid values for this entry are as follows:

`server_name`
The Security Verify Access authorization server name for the WebSEAL server. This name is the name of the authorization API administration server that is used in the **`server task`** commands.

`client-ip-v4`
The IPv4 address of the client of this request.

`client-ip-v6`
The IPv6 address of the client of this request.

`client-port`
The port that is used by the client of this request. This port is the client source port and not the destination port.

`host-name`
The host name of the WebSEAL server. WebSEAL obtains this host name from the **web-host-name** configuration entry in the **[server]** stanza if specified. Otherwise, WebSEAL returns the host name of the server itself.

httphdr{*<name>*}
An HTTP header from the request as specified by the *<name>* field. If the HTTP header is not found in the request, WebSEAL uses the value in the [server] tag-value-missing-attr-tag configuration entry as the value for the header.

text{<value>}
The literal value which is to be assigned to the specified header.

credattr{<name>}
An attribute from the user's credential, as specified by the <name> field. If the specified attribute does not exist, the value contained within the [server] tag-value-missing-attr-tag configuration entry will be used as the value for the header. If the specified attribute is a multi-valued attribute the values will be added to a single HTTP header, with each value separated by a comma.

Usage

This stanza entry is optional.

Default Value

None

Example

```
http-header = X-Forwarded-For:client-ip-v4
http-header = X-Forwarded-Host:httphdr{host}
http-header = X-Forwarded-Server:host-name
http-header = X-Deployment-Status:text{green}
http-header = X-Principal:credattr{AZN_CRED_PRINCIPAL_NAME}
```

introspection-endpoint

Use the `introspection-endpoint` to define the introspection endpoint.

Syntax

```
introspection-endpoint = URL
```

Description

This is the introspection endpoint which is called to handle the token introspection. This configuration item may be customized for a particular junction by adding the adjusted configuration item to a `[oauth-introspection:{jct_id}]` stanza, where `'{jct-id}'` refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

Option

URL
The URL of the introspection endpoint.

Usage

This entry is required.

Default value

Example

```
instrospection-endpoint = https://www.ibm.com/oauth/introspect
```

introspection-response-attributes

Use this entry to control which attributes from the response are added as attributes to the credential.

Syntax

```
introspection-response-attribute = [+|-]<json-data> {[+|-]<json-data> ...}
```

Description

Multiple rules can be specified as a space separated list. When an introspection response is received each piece of JSON data will be evaluated against each rule in sequence until a match is found. The corresponding code (+|-) is then used to determine whether the JSON data is added to the credential or not. If the JSON data name does not match a configured rule it is by default added to the credential. This configuration item may be customized for a particular junction by adding the adjusted configuration item to a `[oauth-introspection:{jct_id}]` stanza, where *'{jct-id}'* refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

Usage

This stanza entry is optional.

Example

```
introspection-response-attributes = +scope +client_id +iat +exp
```

mapped-identity

Use this entry to set a formatted string that is used to construct the IBM Security Verify Access principal name from elements of the introspection response.

Syntax

```
mapped-identity = string
```

Description

Claims can be added to the identity string, surrounded by '{}'.

For example, `{username}` constructs a principal name using the `username` claim from the response.

This configuration item may be customized for a particular junction by adding the adjusted configuration item to a `[oauth-introspection:{jct_id}]` stanza, where *'{jct-id}'* refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

Options

string
A formatted string that is used to construct the Security Verify Access principal name from elements of the introspection response.

Usage

This stanza entry is required.

Default value

None.

Example

```
mapped-identity = {username}
```

multivalue-scope

Use the multivalue-scope stanza entry to specify whether the OAuth scopes are stored as multi-value credential attributes.

Syntax

```
multivalue-scope = {true | false}
```

Description

The **multivalue-scope** controls whether the scopes contained in the introspection response are stored as a single attribute or as a multi-valued attribute in the credential. This configuration item may be customized for a particular junction by adding the adjusted configuration item to a `[oauth-introspection:{jct_id}]` stanza, where *'{jct-id}'* refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

Options

true
Scopes are added as a multi-valued attribute.

false
Scopes are added as a single credential attribute.

Usage

This stanza entry is optional.

Default value

The default value is true.

Example

```
multivalue-scope = false
```

oauth-introspection-auth

Enable authentication using an OAuth introspection endpoint.

Syntax

```
oauth-introspection-auth = {none|http|https|both}
```

Description

The OAuth authentication mechanism allows authentication into IBM Security Verify Access where access can be established based on the Bearer token in a Authorization Header (for example, `Authorization = Bearer <Access Token>`). This is relevant for native mobile applications, single page applications (SPAs), and API access authenticated with OAuth.

This configuration item may be customized for a particular junction by adding the adjusted configuration item to a `[oauth-introspection:{jct_id}]` stanza, where `'{jct-id}'` refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

Options

{none|http|https|both}
Specifies which protocols are supported. The value both means both HTTP and HTTPS.

Usage

This entry is required to enable authentication with the OAuth mechanism.

Default value

None

Example

```
oauth-introspection-auth = both
```

proxy

Use this entry to set the proxy that is used to reach the introspection endpoint.

Syntax

```
proxy = URL
```

Description

The configuration entry is defined in URL format: `http[s]://<address>:<port>`. This configuration item may be customized for a particular junction by adding the adjusted configuration item to a `[oauth-introspection:{jct_id}]` stanza, where `'{jct-id}'` refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

Options

URL
The URL of the proxy that is used to reach the introspection endpoint.

Usage

This stanza entry is optional.

Default value

None.

Example

```
proxy = https://example.com:8080
```

token-type-hint

A hint about the type of token that is submitted for introspection.

Syntax

```
token-type-hint = token_type_hint
```

Description

Specifies a hint about the type of token that is being submitted for introspection. This value is passed in the introspection request in the **token_type_hint** field.

This configuration item may be customized for a particular junction by adding the adjusted configuration item to a `[oauth-introspection:{jct_id}]` stanza, where `'{jct-id}'` refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

Options

token_type_hint
A hint which can be used by the server to determine the type of token being supplied.

Usage

This stanza is optional.

Default value

None.

Example

```
token-type-hint = access-token
```

[oauth-introspection:<jct-id>] stanza

auth-method

The introspection request can be authenticated using Basic Authentication or Forms.

Syntax

```
auth-method = {client_secret_post | client_secret_basic}
```

Description

Controls the method by which the authentication information is supplied to the introspection endpoint. This configuration item may be customized for a particular junction by adding the adjusted configuration item to a `[oauth-introspection:{jct_id}]` stanza, where `'{jct-id}'` refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

Options

client_secret_post
The authentication information is supplied in a form POST.

client_secret_basic
The authentication information is provided in an authorization header.

Usage

This stanza entry is required.

Default value

client_secret_post

Example

```
auth-method = client_secret_basic
```

client-id

Use the client identifier to specify the client which will be used when authenticating to the introspection endpoint.

Syntax

```
client-id = string
```

Description

The client identifier that is used to authenticate to the introspection endpoint. This configuration item may be customized for a particular junction by adding the adjusted configuration item to a [oauth-introspection:*{jct_id}*] stanza, where *'{jct-id}'* refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

Options

string
The client identifier that is used to authenticate to the introspection endpoint.

Usage

This stanza is optional.

Default value

None.

Example

```
client-id = 56879ef20e75817b329576
```

client-id-hdr

Use the client identifier header to specify the name of the HTTP header to be used when authenticating to the introspection endpoint.

Syntax

```
client-id-hdr = header-name
```

Description

The name of the HTTP header that contains the client identifier that is used to authenticate to the introspection endpoint. This configuration item may be customized for a particular junction by adding the adjusted configuration item to a `[oauth-introspection:{jct_id}]` stanza, where `'{jct-id}'` refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

Options

header-name
The name of the header that contains the client identifier.

Usage

This configuration entry is mutually exclusive with the `client-id` configuration entry. If the `client-id` configuration entry is provided, this configuration entry is ignored.

Default value

None.

Example

```
client-id-hdr = x-IBM-CLIENT-ID
```

client-secret

Use the client-secret to authenticate to the introspection endpoint.

Syntax

```
client-secret = string
```

Description

The client secret that is used to authenticate to the introspection endpoint. If a client-id field is not configured the secret is treated as a bearer token, otherwise it is used in a basic authentication header. This configuration item may be customized for a particular junction by adding the adjusted configuration item to a `[oauth-introspection:{jct_id}]` stanza, where `'{jct-id}'` refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

Options

string
The client secret that is used to authenticate to the introspection endpoint.

Usage

This stanza entry is required.

Default value

None.

Example

```
client-secret = as5dgi92ytapsejguas
```

continue-on-auth-failure

Use the **continue-on-auth-failure** stanza entry to define whether to continue processing the request if authentication fails.

Syntax

```
continue-on-auth-failure = {yes|true|no|false}
```

Description

This stanza entry controls whether to continue processing the request and try additional authentication mechanisms if the introspection has failed. This configuration item may be customized for a particular junction by adding the adjusted configuration item to a `[oauth-introspection:{jct_id}]` stanza, where *'{jct-id}'* refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

Options

yes|true
Continue processing the request and try additional authentication mechanisms.

no|false
Do not continue processing the request on an introspection failure.

Usage

This stanza entry is optional.

Default value

`true`

Example

```
continue-on-auth-failure = false
```

external-user

Use this entry to set whether the mapped identity should correspond to a known Security Verify Access identity.

Syntax

```
external-user = {true | false}
```

Description

This boolean is used to indicate whether the mapped identity should correspond to a known Security Verify Access identity. When you are creating a session with an external user, the credential does not contain any group membership. The scopes contained in the credential should instead be used for authorization policy. This configuration item may be customized for a particular junction by adding the adjusted configuration item to a `[oauth-introspection:{jct_id}]` stanza, where *'{jct-id}'* refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

Options

true
The mapped identity should correspond to a known Security Verify Access identity.

false
The mapped identity does not need to correspond to a known Security Verify Access identity.

Usage

This stanza entry is required.

Example

```
external-user = true
```

http-header

Use the `http-header` stanza entry to add HTTP headers to the OAuth introspection request.

Syntax

```
http-header = <header-name>:<header-data>
```

Description

Controls the addition of HTTP headers into the OAuth introspection request.

This configuration item may be customized for a particular junction by adding the adjusted configuration item to a [oauth-introspection:*{jct_id}*] stanza, where '*{jct-id}*' refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

Multiple headers can be specified by including this configuration entry multiple times.

Options

<header-name>
The name of the HTTP header that holds the data. Valid strings are limited to the following characters: A-Z, a-z, 0–9, hyphen (-), or underscore (_).

<header-data>
The type of data that WebSEAL adds to the `<header-name>` header of the request. The valid values for this entry are as follows:

`server_name`
The Security Verify Access authorization server name for the WebSEAL server. This name is the name of the authorization API administration server that is used in the **`server task`** commands.

`client-ip-v4`
The IPv4 address of the client of this request.

client-ip-v6
The IPv6 address of the client of this request.

client-port
The port that is used by the client of this request. This port is the client source port and not the destination port.

host-name
The host name of the WebSEAL server. WebSEAL obtains this host name from the **web-host-name** configuration entry in the **[server]** stanza if specified. Otherwise, WebSEAL returns the host name of the server itself.

httphdr{*<name>*}
An HTTP header from the request as specified by the *<name>* field. If the HTTP header is not found in the request, WebSEAL uses the value in the [server] tag-value-missing-attr-tag configuration entry as the value for the header.

text{<value>}
The literal value which is to be assigned to the specified header.

credattr{<name>}
An attribute from the user's credential, as specified by the <name> field. If the specified attribute does not exist, the value contained within the [server] tag-value-missing-attr-tag configuration entry will be used as the value for the header. If the specified attribute is a multi-valued attribute the values will be added to a single HTTP header, with each value separated by a comma.

Usage

This stanza entry is optional.

Default Value

None

Example

```
http-header = X-Forwarded-For:client-ip-v4
http-header = X-Forwarded-Host:httphdr{host}
http-header = X-Forwarded-Server:host-name
http-header = X-Deployment-Status:text{green}
http-header = X-Principal:credattr{AZN_CRED_PRINCIPAL_NAME}
```

introspection-endpoint

Use the `introspection-endpoint` to define the introspection endpoint.

Syntax

```
introspection-endpoint = URL
```

Description

This is the introspection endpoint which is called to handle the token introspection. This configuration item may be customized for a particular junction by adding the adjusted configuration item to a `[oauth-introspection:{jct_id}]` stanza, where `'{jct-id}'` refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

Option

URL
The URL of the introspection endpoint.

Usage

This entry is required.

Default value

Example

```
instrospection-endpoint = https://www.ibm.com/oauth/introspect
```

introspection-response-attributes

Use this entry to control which attributes from the response are added as attributes to the credential.

Syntax

```
introspection-response-attribute = [+|-]<json-data> {[+|-]<json-data> ...}
```

Description

Multiple rules can be specified as a space separated list. When an introspection response is received each piece of JSON data will be evaluated against each rule in sequence until a match is found. The corresponding code (+|-) is then used to determine whether the JSON data is added to the credential or not. If the JSON data name does not match a configured rule it is by default added to the credential. This configuration item may be customized for a particular junction by adding the adjusted configuration item to a `[oauth-introspection:{jct_id}]` stanza, where `'{jct-id}'` refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

Usage

This stanza entry is optional.

Example

```
introspection-response-attributes = +scope +client_id +iat +exp
```

mapped-identity

Use this entry to set a formatted string that is used to construct the IBM Security Verify Access principal name from elements of the introspection response.

Syntax

```
mapped-identity = string
```

Description

Claims can be added to the identity string, surrounded by '{}'.

For example, `{username}` constructs a principal name using the `username` claim from the response.

This configuration item may be customized for a particular junction by adding the adjusted configuration item to a `[oauth-introspection:{jct_id}]` stanza, where `'{jct-id}'` refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

Options

string
A formatted string that is used to construct the Security Verify Access principal name from elements of the introspection response.

Usage

This stanza entry is required.

Default value

None.

Example

```
mapped-identity = {username}
```

multivalue-scope

Use the multivalue-scope stanza entry to specify whether the OAuth scopes are stored as multi-value credential attributes.

Syntax

```
multivalue-scope = {true | false}
```

Description

The **multivalue-scope** controls whether the scopes contained in the introspection response are stored as a single attribute or as a multi-valued attribute in the credential. This configuration item may be customized for a particular junction by adding the adjusted configuration item to a `[oauth-introspection:{jct_id}]` stanza, where *'{jct-id}'* refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

Options

true
Scopes are added as a multi-valued attribute.

false
Scopes are added as a single credential attribute.

Usage

This stanza entry is optional.

Default value

The default value is true.

Example

```
multivalue-scope = false
```

oauth-introspection-auth

Enable authentication using an OAuth introspection endpoint.

Syntax

```
oauth-introspection-auth = {none|http|https|both}
```

Description

The OAuth authentication mechanism allows authentication into IBM Security Verify Access where access can be established based on the Bearer token in a Authorization Header (for example, `Authorization = Bearer <Access Token>`). This is relevant for native mobile applications, single page applications (SPAs), and API access authenticated with OAuth.

This configuration item may be customized for a particular junction by adding the adjusted configuration item to a `[oauth-introspection:{jct_id}]` stanza, where `'{jct-id}'` refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

Options

{none|http|https|both}
Specifies which protocols are supported. The value both means both HTTP and HTTPS.

Usage

This entry is required to enable authentication with the OAuth mechanism.

Default value

None

Example

```
oauth-introspection-auth = both
```

proxy

Use this entry to set the proxy that is used to reach the introspection endpoint.

Syntax

```
proxy = URL
```

Description

The configuration entry is defined in URL format: `http[s]://<address>:<port>`. This configuration item may be customized for a particular junction by adding the adjusted configuration item to a `[oauth-introspection:{jct_id}]` stanza, where `'{jct-id}'` refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

Options

URL
The URL of the proxy that is used to reach the introspection endpoint.

Usage

This stanza entry is optional.

Default value

None.

Example

```
proxy = https://example.com:8080
```

token-type-hint

A hint about the type of token that is submitted for introspection.

Syntax

```
token-type-hint = token_type_hint
```

Description

Specifies a hint about the type of token that is being submitted for introspection. This value is passed in the introspection request in the **token_type_hint** field.

This configuration item may be customized for a particular junction by adding the adjusted configuration item to a `[oauth-introspection:{jct_id}]` stanza, where `'{jct-id}'` refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

Options

token_type_hint
A hint which can be used by the server to determine the type of token being supplied.

Usage

This stanza is optional.

Default value

None.

Example

```
token-type-hint = access-token
```

[oidc] stanza

This stanza contains the settings for OIDC.

oidc-auth

Enable authentication using the OIDC RP mechanism.

Syntax

```
oidc-auth = {https | none}
```

Description

Use this stanza entry to enable authentication using the OIDC RP mechanism.

Options

https
Enable HTTPS authentication.

none
Do not enable authentication.

Usage

This stanza entry is optional.

Default value

none

Example

```
oidc-auth = none
```

default-op

Use this stanza to set the default OP to be used.

Syntax

```
default-op = default
```

Description

This value is used to define which OP should be used if no OP is supplied in the authentication request. It should correspond to the second component of the name of a qualified OIDC configuration stanza. For example, if the configuration stanza was called `'[oidc:default]'`, this entry would correspond to `'default'`.

Options

default
The ID of the default OP to be used.

Usage

This stanza entry is required.

Default value

None.

Example

```
default-op = default
```

[oidc:default] stanza

discovery-endpoint

Use this entry to set the discovery endpoint for the OP.

Syntax

```
discovery-endpoint = endpoint
```

Description

The CA certificate for the discovery endpoint and corresponding authorization and token endpoints must be added to the WebSEAL junction key database.

Options

endpoint
The discovery endpoint for the OP.

Usage

This stanza entry is required.

Default value

None.

Example

```
discovery-endpoint = https://accounts.google.com/.well-known/openid-configuration
```

redirect-uri-host

Use this entry to set the redirect URI that has been registered with the OIDC OP.

Syntax

```
redirect-uri-host = uri
```

Description

This is the host which is used in the redirect URI registered with the OIDC OP. If no redirect URI host is configured, the host header from the request will be used.

Options

uri
The host which is used in the redirect URI registered with the OIDC OP. The format of the redirect URI is: `https://<host>/pkmsoidc`.

Usage

This stanza entry is optional.

Default value

None.

Example

```
redirect-uri-host = webseal.ibm.com
```

proxy

Use this entry to set the proxy that is used to reach the OIDC endpoints.

Syntax

```
proxy = URL
```

Description

The configuration entry should be in URL format (i.e. `http[s]:<address>{:<port>}`)

Options

URL
The URL of the proxy that is used to reach the OIDC endpoints.

Usage

This stanza entry is optional.

Default value

None.

Example

```
proxy = https://example.com:8080
```

client-identity

Use this entry to set the Security Verify Access client identity as registered with the OP.

Syntax

```
client-identity = string
```

Description

The Security Verify Access client identity as registered with the OP.

Options

string
The Security Verify Access client identity as registered with the OP.

Usage

This stanza entry is required.

Default value

None.

Example

```
client-identity = 56879ef20e75817b329576
```

client-secret

Use this entry to set the Security Verify Access client secret as registered with the OP.

Syntax

```
client-secret = string
```

Description

The Security Verify Access client secret as registered with the OP.

Options

string
The Security Verify Access client secret as registered with the OP.

Usage

This stanza entry is required.

Default value

None.

Example

```
client-secret = as5dgi92ytapsejguas
```

response-type

The required response type for authentication responses.

Syntax

```
response-type = {code | id_token | id_token token}
```

Description

Use this entry to set the required response type for authentication responses.

Options

code
The authorization code flow will be used to retrieve both an access token and identity token.

id_token
The implicit flow will be used to retrieve the identity token.

id_token token
The implicit flow will be used to retrieve both an access token and identity token.

Usage

This stanza entry is required.

Default value

code

Example

```
response-type = code
```

enable-pkce

Use this entry to enable Proof Key for Code Exchange (RFC 7636) during the Authorization Code Flow. Enable this option if the configured OIDC OP requires PKCE.

Syntax

```
enable-pkce = {true | false}
```

Description

This boolean is used to indicate whether the reverse proxy uses Proof Key for Code Exchange (RFC 7636) during the Authorization Code Flow.

Options

true
The reverse proxy uses PKCE during the Authorization Code Flow.

false
The reverse proxy does not use PKCE during the Authorization Code Flow.

Usage

This stanza entry is optional.

Default value

false

Example

```
enable-pkce = true
```

response-mode

Use this entry to set the required response mode for authentication responses.

Syntax

```
response-mode = mode
```

Description

If no response mode is configured, the response mode parameter will not be included in the authentication request.

Options

mode
The possible values include **query**, **fragment**, and **form_post**.

Usage

This stanza entry is optional.

Default value

None.

Example

```
response-mode = query
```

scopes

Use this entry to set the scopes to be sent in the authentication request in addition to the **openid** scope.

Syntax

```
scopes = scope
```

Description

Multiple scopes should be separated by a space.

Options

scope
Possible values include **profile**, **email**, **address**, and **phone**.

Usage

This stanza entry is optional.

Default value

None.

Example

```
scopes = profile email
```

bearer-token-attributes

Use this entry to set a JSON data element from the bearer token response, which should be included in the credential as an extended attribute.

Syntax

```
bearer-token-attributes = [+|-]<json-data>
bearer-token-attributes = <json-data>:<name>
```

Description

Multiple rules can be specified by creating additional configuration entries of the same name. When a bearer token is received each JSON data element will be evaluated against each rule in sequence until a match is found. The corresponding code (+|-) will then be used to determine whether the JSON data will be added to the credential or not. If the JSON data name does not match a configured rule it will by default be added to the credential.

Options

1. For including or excluding JSON data as credential attributes:

   [+|-]<json-data>

   +
   Indicates that this JSON data should be added to the credential.

   -
   Indicates that this JSON data should not be added to the credential.

   <json-data>
   The corresponding JSON data name, which can also contain pattern matching characters (i.e. * ?).

2. For mapping JSON data to other credential attributes:

   <json-data>:<name>

   <json-data>
   The name of the claim.

   <name>
   The name of the credential attribute the claim is mapped to.

Usage

This stanza entry is optional.

Default value

None.

Example

```
bearer-token-attributes = -access_token
bearer-token-attributes = expires_in:bearer_token_expiry
```

id-token-attributes

Use this entry to set a claim from the Id token response which should be included in the credential as an extended attribute.

Syntax

```
id-token-attributes = [+|-]<claim>
```

Description

Multiple rules can be specified by creating additional configuration entries of the same name. When an Id token is received each claim will be evaluated against each rule in sequence until a match is found. The corresponding code (+|-) will then be used to determine whether the claim will be added to the credential or not. If the claim name does not match a configured rule it will by default be added to the credential.

Options

1. For including or excluding claims as credential attributes:

   [+|-]<claim>

   +
   Indicates that this claim should be added to the credential.

   -
   Indicates that this claim should not be added to the credential.

   <claim>
   The corresponding claim name, which can also contain pattern matching characters (i.e. * ?).

2. For mapping claims to other credential attributes:

   <claim>:<name>

   <claim>
   The name of the claim.

   <name>
   The name of the credential attribute the claim is mapped to.

Usage

This stanza entry is optional.

Default value

None.

Example

```
id-token-attributes = = -email
id-token-attributes = auth_time:AZN_CRED_AUTH_TIME
```

allowed-query-arg

Additional query string arguments can be provided to the authentication kick-off URL which will in turn be appended to the corresponding authentication request.

Syntax

```
allowed-query-arg = argument
```

Description

This configuration entry is used to define an allowed query string argument. Any other arguments passed to the kick-off URL will be ignored. This configuration entry can be specified multiple times, once for each allowable query string argument.

Options

argument
The argument to be appended to the corresponding authentication request.

Usage

This stanza entry is optional.

Default value

None.

Example

```
allowed-query-arg = field1
```

```
allowed-query-arg = prompt
allowed-query-arg = max_age
```

```
<meta http-equiv="refresh" content="0;URL='<http://<mycompany.com/pkmsoidc?
iss=i<issuer>&max_age=0&prompt=login'> />
```

mapped-identity

Use this entry to set a formatted string that is used to construct the Security Verify Access principal name from elements of the ID token.

Syntax

```
mapped-identity = string
```

Description

Claims can be added to the identity string, surrounded by '{}'. For example, `{iss}/{sub}` would construct a principal name like the following:

`https://server.example.com/248289761001`

Options

string
A formatted string that is used to construct the Security Verify Access principal name from elements of the ID token.

Usage

This stanza entry is optional.

Default value

None.

Example

```
mapped-identity = {iss}/{sub}
```

external-user

Use this entry to set whether the mapped identity should correspond to a known Security Verify Access identity.

Syntax

```
external-user = {true | false}
```

Description

This boolean is used to indicate whether the mapped identity should correspond to a known Security Verify Access identity.

Options

true
The mapped identity should correspond to a known Security Verify Access identity.

false
The mapped identity does not need to correspond to a known Security Verify Access identity.

Usage

This stanza entry is required.

Default value

`true`

Example

```
external-user = true
```

[obligations-levels-mapping] stanza

obligation

Syntax

```
<obligation> = <authentication-level>
```

Description

Defines the mappings between the obligation levels that the policy decision point (PDP) returns and the WebSEAL step-up authentication levels. Include a separate entry for each obligation that runtime security services (RTSS) returns to the runtime security services EAS.

The mapping between the obligation levels and the WebSEAL authentication levels must be one-to-one. The user must authenticate only through the appropriate obligation mechanisms.

The runtime security services EAS maps the obligation to the authentication level specified in this stanza and requests WebSEAL to authenticate the user at that level.

Options

<obligation>
The name of the obligation that RTSS returns to the runtime security services EAS.

<authentication-level>
The WebSEAL authentication level that the runtime security services EAS includes in the WebSEAL request. This value is a number that represents the authentication level in the **[authentication-levels]** stanza. Each entry in the **[authentication-levels]** is assigned a number based on its position in the list; the first entry is level `0`. For more information, see the *IBM Security Verify Access: Web Reverse Proxy Configuration Guide* and search for *specifying authentication levels*.

Usage

This stanza entry is required.

Default value

None.

Example

```
life_questions=2
otp=3
email=4
voice=5
```

[obligations-urls-mapping] stanza

Use this stanza to define a URL to be used to satisfy an obligation.

Note: This stanza is not included in the WebSEAL configuration file by default. You can manually add this stanza and the associated entries if you want to configure runtime security services for Advanced Access Control.

obligation

Define a URL to be used to satisfy an obligation.

Syntax

```
obligation = URL
```

Description

This entry defines the mapping between the obligation that the policy decision point (PDP) returns and the URL that is used to satisfy the obligation.

Options

obligation
Defines the obligation that is returned by runtime security services. This string, or *key*, is case-sensitive.

URL
Defines the URL that is used to satisfy the obligation.

Usage

This stanza entry is not required.

The **[obligations-urls-mapping]** stanza and the *obligation* entry apply only to Advanced Access Control.

Examples

The following entry specifies an obligation named `auth1`. The value of `auth1` is a URL that is used to satisfy the obligation.

```
[obligations-urls-mapping]
auth1 = https://example.com
```

[p3p-header] stanza

access

Use the **access** stanza entry to specify the type of cookie information that a user can access.

Syntax

```
access = {none|all|nonident|contact-and-other|ident-contact|other-ident}
```

Description

Specifies the user access to the information contained in and linked to the cookie.

Options

none
No access to identified data is given.

all
Access is given to all identified data.

contact-and-other
Access is given to identified online and physical contact information, in addition to certain other identified data.

ident-contact
Access is given to identified online and physical contact information. For example, users can access things such as a postal address.

nonident
Website does not collect identified data.

other-ident
Access is given to certain other identified data. For example, users can access things such as their online account charges.

Usage

This stanza entry is required.

Default value

none

Example

```
access = none
```

categories

Use the **categories** stanza entry to specify the type of information that is stored in the cookie or linked to by the cookie.

Syntax

```
categories = {physical|online|uniqueid|purchase|financial|computer|
navigation|interactive|demographic|content|state|political|health|
preference|location|government|other-category}
```

Description

Specifies the type of information stored in the cookie or linked to by the cookie. When the **non-identifiable** stanza entry is set to yes, then no categories need be configured.

Options

physical
Information that allows an individual to be contacted or located in the physical world. For example, telephone number or address.

online
Information that allows an individual to be contacted or located on the Internet.

uniqueid
Non-financial identifiers, excluding government-issued identifiers, issued for purposes of consistently identifying or recognizing the individual.

purchase
Information actively generated by the purchase of a product or service, including information about the method of payment.

financial
Information about an individual's finances including account status and activity information such as account balance, payment or overdraft history, and information about an individual's purchase or use of financial instruments including credit or debit card information.

computer
Information about the computer system that the individual is using to access the network. For example, IP number, domain name, browser type or operating system.

navigation
Data *passively* generated by *browsing* the Web site. For example, which pages are visited, and how long users stay on each page.

interactive
Data *actively* generated from or reflecting *explicit interactions* with a service provider through its site. For example, queries to a search engine, or logs of account activity.

demographic
Data about an individual's characteristics. For example, gender, age, and income.

content
The words and expressions contained in the body of a communication. For example, the text of email, bulletin board postings, or chat room communications.

state
Mechanisms for maintaining a stateful session with a user or automatically recognizing users who have visited a particular site or accessed particular content previously. For example, HTTP cookies.

political
Membership in or affiliation with groups such as religious organizations, trade unions, professional associations and political parties.

health
Information about an individual's physical or mental health, sexual orientation, use or inquiry into health care services or products, and purchase of health care services or products

preference
Data about an individual's likes and dislikes. For example, favorite color or musical tastes.

location
Information that can be used to identify an individual's current physical location and track them as their location changes. For example, Global Positioning System position data.

government
Identifiers issued by a government for purposes of consistently identifying the individual.

other-category
Other types of data not captured by the above definitions.

Usage

This stanza entry is required.

Default value

`uniqueid`

Example

```
categories = uniqueid
```

disputes

Use the **disputes** stanza entry to specify whether information about disputes is contained in the P3P policy.

Syntax

```
disputes = {yes|no}
```

Description

Specifies whether the full P3P policy contains some information regarding disputes over the information contained within the cookie.

Options

yes
The value `yes` means that information about disputes is contained in the full P3P policy.

no
The value `no` means that no information about disputes is contained in the policy.

Usage

This stanza entry is required.

Default value

no

Example

```
disputes = no
```

enable-p3p

Use the `enable-p3p` configuration entry to enable or disable the insertion of the P3P header into the HTTP response.

Syntax

```
enable-p3p = {yes|no}
```

Description

Whether the P3P header, defined by the policy contained within the `[p3p-header]` configuration stanza, will be added to the response.

If this configuration entry is set to no, and the "preserve-p3p-policy" on page 436 configuration entry within the [server] stanza is set to no, any P3P headers which have been set by the junction server will be removed from the response.

Options

yes
The P3P header will be added to the response.

no

The P3P header will not be added to the response.

Usage

The stanza entry is optional.

Default value

`yes`

Example

```
enable-p3p = yes
```

non-identifiable

Syntax

```
non-identifiable = {yes|no}
```

Description

Specifies that no information in the cookie, or linked to by the cookie, personally identifies the user.

Options

yes
Data that is collected identifies the user.

no
No data is collected (including Web logs), or the information collected does not identify the user.

Usage

This stanza entry is required.

Default value

no

Example

```
non-identifiable = no
```

p3p-element

Syntax

```
p3p-element = policyref=location_of_policy_reference
```

Description

Specifies elements to add to the P3P header in addition to the elements specified by the other configuration items in this stanza. Typically this is done by referring to the location of a full XML policy.

Options

policyref=*location_of_policy_reference*
The default entry points to a default policy reference located on the World Wide Web Consortium Web site.

Usage

This stanza entry is required.

Default value

The default entry points to a default policy reference located on the World Wide Web Consortium Web site.

```
policyref="/w3c/p3p.xml"
```

Example

```
p3p-element = policyref="/w3c/p3p.xml"
```

purpose

Syntax

```
purpose = {current|admin|develop|tailoring|pseudo-analysis|
pseudo-decision|individual-analysis|individual-decision|
contact|historical|telemarketing|other-purpose}
[:[opt-in|opt-out|always]]
```

Description

Specifies the purpose of the information in the cookie and linked to by the cookie.

Options

current
Information can be used by the service provider to complete the activity for which it was provided.

admin
Information can be used for the technical support of the Web site and its computer system.

develop
Information can be used to enhance, evaluate, or otherwise review the site, service, product, or market.

tailoring
Information can be used to tailor or modify content or design of the site where the information is used only for a single visit to the site.

pseudo-analysis
Information can be used to create or build a record of a particular individual or computer that is tied to a pseudonymous identifier, without tying identified data (such as name, address, phone number, or email address) to the record. This profile will be used to determine the habits, interests, or other characteristics of individuals *for purpose of research, analysis and reporting*, but it will not be used to attempt to identify specific individuals.

pseudo-decision
Information can be used to create or build a record of a particular individual or computer that is tied to a pseudonymous identifier, without tying identified data (such as name, address, phone number, or email address) to the record. This profile will be used to determine the habits, interests, or other characteristics of individuals *to make a decision that directly affects that individual*, but it will not be used to attempt to identify specific individuals.

individual-analysis
Information can be used to determine the habits, interests, or other characteristics of individuals and combine it with identified data *for the purpose of research, analysis and reporting*.

individual-decision
Information can be used to determine the habits, interests, or other characteristics of individuals and combine it with identified data *to make a decision that directly affects that individual*.

contact
Information can be used to contact the individual, through a communications channel other than voice telephone, for the promotion of a product or service.

historical
Information can be archived or stored for the purpose of preserving social history as governed by an existing law or policy.

telemarketing
Information can be used to contact the individual though a voice telephone call for promotion of a product or service.

other-purpose
Information may be used in other ways not captured by the above definitions.

For all values except `current`, an additional option can be specified. The possible values are:

always
Users cannot opt-in or opt-out of this use of their data.

opt-in
Data may be used for this purpose only when the user affirmatively requests this use.

opt-out
Data may be used for this purpose unless the user requests that it not be used in this way.

When no additional option is specified, the default value is `always`.

Usage

This stanza entry is required.

Default value

The default values are `current` and `other-purpose:opt-in`.

Example

```
purpose = current
purpose = other-purpose:opt-in
```

recipient

Syntax

```
recipient = {ours|delivery|same|unrelated|public|
other-recipient}[:[opt-in|opt-out|always]]
```

Description

Specifies the recipients of the information in the cookie, and linked to by the cookie.

Options

ours
Ourselves and/or entities acting as our agents, or entities for whom we are acting as an agent. An *agent* is a third party that processes data only on behalf of the service provider.

delivery
Legal entities *performing delivery services* that may use data for purposes other than completion of the stated purpose.

same
Legal entities following our practices. These are legal entities who use the data on their own behalf under equable practices.

unrelated
Unrelated third parties. These are legal entities whose data usage practices are not known by the original service provider.

public
Public forums. These are public forums such as bulletin boards, public directories, or commercial CD-ROM directories.

other-recipient
Legal entities following different practices. These are legal entities that are constrained by and accountable to the original service provider, but may use the data in a way not specified in the service provider's practices.

For all values an additional option can be specified. The possible values are:

always
Users cannot opt-in or opt-out of this use of their data.

opt-in
Data may be used for this purpose only when the user affirmatively requests this use.

opt-out
Data may be used for this purpose unless the user requests that it not be used in this way.

When no additional option is specified, the default value is `always`.

Usage

This stanza entry is required.

Default value

`ours`

Example

```
recipient = ours
recipient = public:opt-in
```

remedies

Syntax

```
remedies = {correct|money|law}
```

Description

Specifies the types of remedies in case a policy breach occurs. When this entry has no value, there is no remedy information in the P3P compact policy.

Options

correct
Errors or wrongful actions arising in connection with the privacy policy will be remedied by the service.

money
If the service provider violates its privacy policy it will pay the individual an amount specified in the human readable privacy policy or the amount of damages.

law
Remedies for breaches of the policy statement will be determined based on the law referenced in the human readable description.

Usage

This stanza entry is required.

Default value

`correct`

Example

```
remedies = correct
```

retention

Syntax

```
retention = {no-retention|stated-purpose|legal-requirement|
business-practices|indefinitely}
```

Description

Specifies how long the information in the cookie or linked to by the cookie is retained.

Options

no-retention
Information is not retained for more than a brief period of time necessary to make use of it during the course of a single online interaction.

stated-purpose
Information is retained to meet the stated purpose, and is to be discarded at the earliest time possible.

legal-requirement
Information is retained to meet a stated purpose, but the retention period is longer because of a legal requirement or liability.

business-practices
Information is retained under a service provider's stated business practices.

indefinitely
Information is retained for an indeterminate period of time.

Usage

This stanza entry is required.

Default value

`no-retention`

Example

```
retention = no-retention
```

[PAM] stanza

pam-enabled

Syntax

```
pam-enabled = {true|false}
```

Description

Enables or disables the IBM Internet Security Systems Protocol Analysis Module. The module inspects the HTTP content of selected requests, checking for potential security vulnerabilities.

Options

true
Enables the Protocol Analysis Module.

false
Disables the Protocol Analysis Module.

Usage

This stanza entry is required.

Default value

`false`

Example

```
pam-enabled = false
```

pam-simulation-mode-enabled

Syntax

```
pam-simulation-mode-enabled = {true|false}
```

Description

Enables or disables the IBM Internet Security Systems Protocol Analysis Module simulation mode. If the simulation mode is enabled, any issues that are detected are audited and then ignored. This mode provides a mechanism for administrators to see what issues are being detected without having an impact on the client traffic.

Options

true
Enables the Protocol Analysis Module simulation mode.

false
Disables the Protocol Analysis Module simulation mode.

Usage

This stanza entry is required.

Default value

`false`

Example

```
pam-simulation-mode-enabled = false
```

pam-max-memory

Syntax

```
pam-max-memory = memory_size
```

Description

The amount of memory, in bytes, that the IBM Internet Security Systems Protocol Analysis Module can use per worker-thread. The module uses this value to tune the size of its caches for the amount of available memory.

Options

memory_size

The amount of memory, in bytes, that is available to the module per worker-thread.

Usage

This stanza entry is optional.

Default value

None.

Example

```
pam-max-memory = 16777216
```

pam-use-proxy-header

Syntax

```
pam-use-proxy-header = {true|false}
```

Description

Controls whether the Protocol Analysis Module uses the **`X-Forwarded-For`** header to identify the client. This configuration item is useful if a network-terminating proxy is located between the server and the client. If the value is set to `false`, the module identifies the client based on the socket connection information.

Options

true

The module uses the **`X-Forwarded-For`** header to identify the client.

false
The module uses the available socket connection information to identify the client.

Usage

This stanza entry is required.

Default value

`false`

Example

```
pam-use-proxy-header = false
```

pam-http-parameter

Syntax

```
pam-http-parameter = parameter:value
```

Description

Defines specific parameters for WebSEAL to pass to the Protocol Analysis Module HTTP interface during initialization. For a list of valid Protocol Analysis Module parameters, see the module documentation at http://www.iss.net/security_center/reference/help/pam.

Note: You can specify this configuration entry multiple times, one for each parameter.

Options

parameter*:*value
The Protocol Analysis Module parameter and its assigned value.

Usage

This stanza entry is optional.

Default value

None.

Example

```
pam-http-parameter = param1:val1
pam-http-parameter = param2:val2
```

pam-coalescer-parameter

Syntax

```
pam-coalescer-parameter = parameter:value
```

Description

Defines specific parameters for WebSEAL to pass to the Protocol Analysis Module coalescer interface during initialization. The Protocol Analysis Module uses this interface to combine module-related issues into a single event. For a list of valid Protocol Analysis Module parameters, see the module documentation at https://pam.xforce-security.com/security_center/reference/help/.

Note: You can specify this configuration entry multiple times, one for each parameter.

Options

parameter*:*value
The Protocol Analysis Module parameter and its assigned value.

Usage

This stanza entry is optional.

Default value

None.

Example

```
pam-coalescer-parameter = combine:on
```

pam-log-cfg

Syntax

```
pam-log-cfg = agent [parameter=value],[parameter=value]...
```

Description

Configures the IBM Internet Security Systems Protocol Analysis Module for logging. You can use the available parameters to configure the logging agents.

Options

agent
Specifies the logging agent. The agent controls the logging destination for server events. Valid agents include:

- `stdout`
- `stderr`
- `file`
- `remote`
- `rsyslog`

parameter
The different agents support the following configuration parameters:

Table 3. Logging agent configuration parameters

Parameter	Supporting agents
`buffer_size`	`remote`

Table 3. Logging agent configuration parameters (continued)	
Parameter	**Supporting agents**
compress	remote
dn	remote
error_retry	remote, rsyslog
flush_interval	all
hi_water	all
log_id	file, rsyslog
max_event_len	rsyslog
mode	file
path	all
port	remote, rsyslog
queue_size	all
rebind_retry	remote, rsyslog
rollover_size	file
server	remote, rsyslog
ssl_keyfile	rsyslog
ssl_label	rsyslog
ssl_stashfile	rsyslog

Note: For a complete description of the available logging agents and the supported configuration parameters, see the *IBM Security Verify Access for Web: Auditing Guide*.

Usage

This stanza entry is required.

Default value

None.

Example

To send logging from the Protocol Analysis Module to a file called pam.log:

```
pam-log-cfg = file path=pam.log
```

To send logging from the module to a remote syslog server:

```
pam-log-cfg = rsyslog server=timelord,port=514,log_id=webseal-instance
```

pam-log-audit-events

Syntax

```
pam-log-audit-events = {true|false}
```

Description

Specifies whether audit events are sent to the Protocol Analysis Module log file.

Note: You can use the **pam-log-cfg** entry in the **[PAM]** stanza to configure the log file for the module.

Options

true
The Protocol Analysis Module sends audit events to the log file.

Note: This setting dramatically increases the number of logged events.

false
The Protocol Analysis Module does not send audit events to the log file.

Usage

This stanza entry is required.

Default value

`false`

Example

```
pam-log-audit-events = false
```

pam-disabled-issues

Syntax

```
pam-disabled-issues = list_of_issues
```

Description

Specifies a comma-separated list of Protocol Analysis Module issues to disable. By default, all Protocol Analysis Module issues are enabled.

Options

list_of_issues
A comma-separated list of Protocol Analysis Module issues. The module disables each issue in the list.

Usage

This stanza entry is optional.

Default value

None.

Example

The following entry disables **Ace_Filename_Overflow** and **HTTPS_Apache_ClearText_DoS**.

```
pam-disabled-issues = 2121050,2114033
```

pam-resource-rule

Syntax

```
pam-resource-rule = [+|-]{URI}
```

Description

Specifies the rules that WebSEAL uses to determine whether to pass a particular resource down to the Protocol Analysis Module. WebSEAL examines each rule in sequence until a match is found. The first successful match determines whether WebSEAL passes the request to the module. WebSEAL does not pass the request to the module layer if no match is found.

You can define multiple resource rules. Each entry has the format: `[+|-]{URI}`. For example, `-*.gif`.

Options

+
Configures WebSEAL to pass matching requests to the Protocol Analysis Module layer.

-
Configures WebSEAL not to pass matching requests to the Protocol Analysis Module layer.

{URI}
Contains a pattern that WebSEAL uses to match against the URI that is found in the request. You can use the wildcard characters * and ?.

Usage

This stanza entry is optional.

Default value

None.

Example

```
pam-resource-rule = -*.gif
pam-resource-rule = +*.html
```

pam-fail-early

Syntax

```
pam-fail-early = {true|false}
```

Description

Enable or disable this entry to control whether blocked requests fail early or late.

Options

True
Any blocked requests fail as soon as an issue is detected.

False
Any blocked requests fail later, allowing additional fields to be added to the audit record.

Usage

Required.

This stanza entry is used to control whether the connection is dropped as soon as an issue is detected. If the connection is dropped early it will have less impact on WebSEAL. However, it also means that the audit record will not contain the user name and certain pieces of information derived from the request. For example, browser information.

Default value

`True`

Example

```
pam-fail-early = true
```

pam-use-epoch-time

Syntax

```
pam-use-epoch-time = {true|false}
```

Description

Enable or disable this option to control the format of the time fields within the generated audit records.

Options

True
The time field in the audit records are represented as the seconds since Epoch.

False
The time field contained in the audit records are in ASCII string format.

Usage

Required.

This stanza entry controls whether the time fields which are included in the audit records are an ASCII string representation of the time or the number of seconds since Epoch.

Default value

`True`

Example

```
pam-use-epoch-time = true
```

[pam-resource:*<URI>*] stanza

You can use this stanza to customize the Protocol Analysis Module processing for individual resources and events. The *<URI>* value contains a pattern that WebSEAL can match against the URI that is found in the request. You can use the wildcard characters `*` and `?`. For example, `[pam-resource:test.html]` or `[pam-resource:*.js]`.

pam-issue

Syntax

```
pam-issue = action
```

Description

You can use the entries in this stanza to control the processing of certain module-related events.

Options

pam-issue
Contains a pattern, which WebSEAL uses to match a Protocol Analysis Module issue. You can use the wildcard characters * and ?.

action
The action to undertake for the issue. The action can be either of the following values:

`block`
Blocks the connection for a specified number of seconds. For example, `block:30`.

`ignore`
Ignores the issue and continues to process the request.

Usage

This stanza entry is required.

If the **pam-resource** stanza does not contain at least a single **pam-issue=action** value, then the stanza will not be replicated to the target instance when a server sync command is issued.

Default value

None.

Example

```
212105? = block:0
2119002 = block:20
```

[password-strength] stanza

Use the **[password-strength]** stanza to define settings for the password strength module.

The password strength module validates the strength of new passwords. This module is invoked by a password change operation. It evaluates the new password against an XSLT rule to determine whether the new password meets the configured criteria.

rules-file

Use the **rules-file** stanza entry to define the name of the rules file to be used by the password strength module.

Syntax

```
rules-file = file-name
```

Description

The name of the rules file that is used by the password strength module.

Options

file-name
Name of the rules file.

Usage

This stanza entry is required.

Default value

None.

Example

```
rules-file = password-rules.txt
```

debug-level

Use the **debug-level** stanza entry to define the initial tracing level of the password strength module.

Syntax

```
debug-level = level
```

Description

Controls the initial trace level for the password strength module.

Options

level
The level variable that indicates the trace level of the password strength module, with 1 designating a minimal amount of tracing and 9 designating the maximum amount of tracing.

Note: You can also use the Security Verify Access **pdadmin** trace commands to modify the trace level by using the trace component name of `pd.cas.pwdstrength`. This trace component is only available after the first password change operation is processed.

Usage

This stanza entry is optional.

Default value

`0`

Note: A debug level of `0` results in no tracing output.

Example

```
debug-level = 0
```

password-callouts stanza

The `password-callouts` stanza is used to configure the system to make a REST call before and after password update operations (`/pkmspasswd`) take place. This, for example, allows external services to perform password strength validation.

authentication-endpoint

Use the authentication endpoint to specify an endpoint that is called to obtain an access token which can then be used in the subsequent password update callouts.

Syntax

`authentication-endpoint = endpoint`

Description

This is the endpoint that is called to obtain an access token which can then be used in the subsequent password update callouts. If no endpoint is specified a Basic Authentication header, constructed from the client-id and client-secret configuration entries, is used. The endpoint should conform to the OAuth client credential flow (OAuth 2.0 RFC 6749, section 4.4).

Options

Endpoint
Specifies the endpoint that is called to obtain an access token which can be used in the subsequent password update callouts.

Usage

This stanza entry is required.

Default value

None.

Example

```
authentication-endpoint = https://www.sample.com/oauth/token
```

client-id

Use this entry to specify the client identifier which is used when authenticating to the callout services.

Syntax

`client-id = client_identifier_value`

Description

The client identifier, that is used when you are authenticating to the callout services.

Options

client_identifier_value
Specifies the client identifier that is used to authenticate to the callout services.

Usage

This stanza entry is optional.

Default Value

None

Example

```
client-id = jsmith
```

client-secret

Use this entry to specify the client password which is used when you are authenticating to the callout services

Syntax

```
client-secret = string
```

Description

The client secret, which is used when authenticating to the callout services.

Options

string
The password of the client.

Usage

This stanza entry is required.

Default Value

None.

Example

```
client-secret = as5dgi92ytapsejguas
```

search-endpoint

Use this entry to specify the endpoint that is called to map the IBM Security Verify Access user identity into an identity which is known to the callout service.

Syntax

```
search-endpoint = endpoint
```

Description

The endpoint which is called to map the IBM Security Verify Access user identity into an identity which is known to the callout services. This endpoint should conform to the System for Cross-domain Identity Management RFC 7644 (section 3.4.2). If no endpoint is specified the IBM Security Verify Access user identity, as provided in the password change request, is used in the callout to the pre and/or post update services.

Options

Endpoint
The endpoint which is used to perform the user search.

Usage

This entry is optional.

Default Value

None.

Example

```
search-endpoint = https://www.ibm.com/scim/users/.search
```

search-filter

Use this entry to specify the filter which is used when mapping the Verify Access user identity into an identity which is known to the callout services.

Syntax

```
search-filter = <filter-string>
```

Description

The search filter which is used when mapping the Verify Access user identity into an identity which is known to the callout services. This search filter should conform to the System for Cross-domain Identity Management RFC 7644 (section 3.4.2). The '{username}' macro can be used in the filter to indicate the user identity which has been provided in the password change request. An example filter can be `userName eq "{username}"`.

Options

`<filter-string>`
Specifies the name of the filter that is used to map the Verify Access user identity into an identity which is known to the callout services.

Usage

This stanza entry is optional.

Example

```
search-filter = username eq "{username}"
```

pre-update-endpoint

Use this entry to specify the endpoint to be called to perform any password pre-update processing.

Syntax

```
pre-update-endpoint = endpoint
```

Description

The endpoint which is called to perform any password pre-update processing. This endpoint should conform to the draft-hunt-scim-password-mgmt-00 RFC (section 2.5).

Options

endpoint
The endpoint which is used to perform any password pre-update processing.

Usage

This stanza entry is required.

Default value

None.

Example

```
pre-update-endpoint = https://www.sample.com/scim/PasswordValidateRequests
```

pre-update-user-prefix

Use this entry to specify the prefix which is used when you are building the user identity to be included in the password pre-update callout.

Syntax

```
pre-update-user-prefix = <prefix string>
```

Description

The prefix which is used when you are building the user identity to be included in the password pre-update callout.

Options

<prefix string>
Specifies the string which will prefix the user identity in the password pre-update callout.

Usage

This stanza entry is optional.

Default value

None.

Example

```
pre-update-user-prefix = /Users/
```

post-update-endpoint

Use this entry to specify the endpoint to be called to perform any password post-update processing.

Syntax

```
post-update-endpoint = endpoint
```

Description

The endpoint which is called to perform any password post-update processing. This endpoint should conform to the System for Cross-domain Identity Management: Protocol RFC 7644 (section 3.5.2).

Options

endpoint
The endpoint which is used to perform any password post-update processing.

Usage

This stanza entry is required.

Default value

None.

Example

```
post-update-endpoint = https://www.sample.com/scim/Users
```

proxy

Use this stanza entry to define the proxy, if any, to reach the various endpoints.

Syntax

`proxy = http[s]://<address>:<port>`

Description

The proxy that is used to reach the various endpoints.

Options

address:port
Specify the host name and port number of the proxy

Usage

This stanza entry is optional.

Default Value

None.

Example

`proxy = https://www.example.com:8080`

static-header

Use this stanza entry to define any static headers which should be added to the callout requests.

Syntax

`static-header = header-name:header-value`

Description

Any static headers which should be added to the callout requests. This configuration entry can be specified multiple times, one for each header that needs to be added to the callout requests.

Options

header-name:header-value
Specifies the header name and the header value to be added to the callout requests.

Usage

This stanza entry is optional.

Default value

None.

Example

```
static-header = realm:ibm.com
```

[preserve-cookie-names] stanza

name

Use the **name** entry in the **[preserve-cookie-names]** stanza to list specific cookie names that WebSEAL must not modify.

Syntax

```
name = cookie_name
```

Description

WebSEAL, by default, modifies the names of cookies returned in responses from junctions created with pdadmin using –j flag. WebSEAL also by default modifies the name of cookies listed in the junction mapping table (JMT). This default modification is done to prevent naming conflicts with cookies returned by other junctions.

When a front-end application depends on the names of specific cookies, the administrator can disable the modification of cookie names for those specific cookies. The administrator does this by listing the cookies in this stanza.

Note: You can configure more than one **name** entry to list multiple cookie names that WebSEAL must not modify.

Options

cookie_name
When entering a value for *cookie_name*, use ASCII characters.

Usage

This stanza entry is optional.

Default value

There are no cookie names set by default.

Example

```
name = JSESSIONID
```

[process-root-filter] stanza

root

Syntax

```
root = pattern
```

Description

Specifies the patterns for which you want root junction requests processed at the root junction when `process-root-requests = filter`.

Options

pattern
Values for *pattern* must be standard WebSEAL wildcard patterns.

Usage

Entries in this stanza are required when `process-root-requests = filter`.

Default value

```
root = /index.html
root = /cgi-bin*
```

Example

```
root = /index.html
root = /cgi-bin*
```

[rate-limiting] stanza

For more information on rate limiting, see 'Rate limiting' in the "Command Reference topics".

policy

Use the `policy` configuration entry to identify the rate limiting policies that are applied to the reverse proxy.

Syntax

```
policy = <Policy name>
```

Description

This configuration entry identifies the rate limiting policies that are applied to the reverse proxy.

Options

<Policy name>
Specifies the name of the policy.

Usage

This stanza entry is optional.

Default Value

None

Example

```
policy = LimitBearerToken
```

redis-enabled

Use the `redis-enabled` configuration entry to control whether the rate-limiting data will be stored in a Redis database

Syntax

```
redis-enabled = <yes|no>
```

Description

This entry controls whether the rate-limiting data will be stored in a Redis database. When stored in a Redis database, rate-limiting data can be shared across multiple WebSEAL instances.

Options

yes
Specifies that the rate-limiting data is stored in a Redis database.

no
Specifies that the rate-limiting data is not stored in a Redis database.

Usage

This stanza entry is optional.

Default Value

no

Example

```
redis-enabled = yes
```

redis-collection-name

Use the `redis-collection-name` configuration entry to specify the name of a Redis collection.

Syntax

```
redis-collection-name = <collection-name>
```

Description

This entry specifies the name of a Redis collection which will be used for maintaining rate-limiting data. If this entry is left blank, the default collection specified in the `[redis]` stanza will be used.

Options

<collection-name>
Specifies the name of the Redis collection.

Usage

This stanza entry is optional.

Default Value

none

Example

```
redis-collection-name = my-redis-collection
```

redis-sync-window

Use the `redis-sync-window` configuration entry to specify the length of time a record from Redis is cached locally.

Syntax

```
redis-sync-window = <number of seconds>
```

Description

The length of time (in seconds) a record from Redis will be cached locally by this instance. Records will only be synchronized with Redis after this window has elapsed.

Smaller values will result in more frequent updates with the Redis database and may have a negative performance impact.

Options

<number of seconds>
Specify the length of time, in seconds.

Usage

This stanza entry is required.

Default Value

5

Example

```
redis-sync-window = 5
```

add-response-headers

Use the `add-response-headers` configuration entry to add rate limiting policy information to the HTTP response.

Syntax

```
add-response-headers = <yes|no>
```

Description

This configuration entry controls whether the rate limiting response headers are added to the response or not. The following HTTP headers are added to the HTTP response for requests that match a rate limiting policy.

X-Rate-Limit-Policy
The name of the rate limiting policy that is closest to being exceeded. The policy name is obtained from the **name** configuration entry within the policy YAML file.

X-Rate-Limit-Remaining
The number of requests that are left for the rate limiting policy in the current rate-limit window.

X-Rate-Limit-Reset
The time (Coordinated Universal Time Epoch time) at which the rate limiting policy resets.

Options

yes
The rate limiting headers are added to HTTP responses.

no
The rate limiting headers are not added to HTTP responses.

Usage

This stanza entry is optional.

Default Value

no

Example

```
add-response-headers = yes
```

[reauthentication] stanza

reauth-at-any-level

Syntax

```
reauth-at-any-level = {yes|no}
```

Description

Controls whether a different authentication level or mechanism is permitted during a reauthentication operation.

Options

yes
During a reauthentication operation, a user can be authenticated using a different authentication level or mechanism from that which is currently held by the user. The user's new credential replaces the old one.

Note: If this configuration option is set to *yes*, the credential can change one or more times during the lifetime of the session. Also, the credential will always be updated upon a successful reauthentication regardless of the existing authentication level of the credential.

no
During a reauthentication operation, a user can only be authenticated at the same authentication level or mechanism as the user's current credential.

Usage

This stanza entry is required.

Default value

no

Example

```
reauth-at-any-level = no
```

reauth-extend-lifetime

Syntax

```
reauth-extend-lifetime = number_of_seconds
```

Description

Integer value expressing the time in seconds that the credential cache timer should be extended to allow clients to complete a reauthentication.

Options

number_of_seconds
When the value is zero (0), the lifetime timer is not extended. WebSEAL imposes no maximum. The maximum value is limited only by the integer data type.

Usage

This stanza entry is required.

Default value

0

Example

```
reauth-extend-lifetime = 0
```

reauth-for-inactive

Syntax

```
reauth-for-inactive = {yes|no}
```

Description

Enables WebSEAL to prompt users to reauthenticate when their entry in the WebSEAL credential cache has timed out due to inactivity.

Options

yes
Enable reauthentication.

no
Disable reauthentication.

Usage

This stanza entry is required.

Default value

no

Example

```
reauth-for-inactive = no
```

reauth-reset-lifetime

Syntax

```
reauth-reset-lifetime = {yes|no}
```

Description

Enables WebSEAL to reset the lifetime timer for WebSEAL credential cache entries following successful reauthentication.

Options

yes
Enable.

no
Disable.

Usage

This stanza entry is required.

Default value

no

Example

```
reauth-reset-lifetime = no
```

terminate-on-reauth-lockout

Syntax

```
terminate-on-reauth-lockout = {yes|no}
```

Description

Specifies whether or not to remove the session cache entry of a user who reaches the **max-login-failures** policy limit during reauthentication.

Options

yes
When the maximum number of failed login attempts (specified by the **max-login-failures** policy) is reached during reauthentication, the user is logged out and the user's session is removed.

no

When the maximum number of failed login attempts (specified by the **max-login-failures** policy) is reached during reauthentication, the user is locked out as specified by the **disable-time-interval** setting, and notified of the lockout as specified by the **late-lockout-notification** setting. The user is not logged out and the initial login session is still valid. The user can still access other resources that are not protected by a **reauthn** POP.

Usage

This stanza entry is required.

Default value

yes

Example

```
terminate-on-reauth-lockout = yes
```

[redis] stanza

Use the `[redis]` stanza to define the Redis servers which can be used to store remote session information.

client-list-cache-lifetime

The client-list-cache-lifetime configuration entry specifies the length of time in seconds that a client list will be cached.

Syntax

```
client-list-cache-lifetime = <lifetime>
```

Description

The WebSEAL server needs to manually delete stale entries from the Redis cache during session creation and idle timeout events. In order to be able to delete the stale entries it needs an up-to-date list of active clients of the Redis server (using the 'CLIENT LIST' Redis command). This command, depending on the number of clients which are registered with the Redis server, can be expensive and so WebSEAL will cache and reuse the returned list of clients for a small period of time. This configuration entry controls the length of time in seconds that a client list will be cached.

Options

<lifetime>
The length of time, in seconds, that a client list will be cached.

Usage

This stanza entry is optional.

Default Value

10

Example

```
client-list-cache-lifetime = 10
```

default-collection-name

The name of the Redis collection of servers which will be used by default.

Syntax

```
default-collection-name = <name>
```

Description

The name of the collection of Redis servers which will be used by default when creating new sessions. The default collection will be used when a request does not match any configured host-specific collections.

Options

<name>
The name of the default collection of Redis servers.

Usage

This stanze entry is required.

Default value

None

Example

```
default-collection-name = collection_a
```

key-prefix

The `key-prefix` configuration entry specifies the prefix which is used for all keys stored in the Redis server.

Syntax

```
key-prefix = <prefix>
```

Description

The key prefix for all data which is stored in the Redis server.

Options

`<prefix>`
The prefix used for all keys.

Usage

This stanza entry is required.

Default value

None

Example

```
key-prefix = isva-
```

[redis-collection:<collection-name>] stanza

Use the `redis-collection:<collection-name>` stanza to define a collection of replicated Redis servers which can be used to store remote session information. The stanza name is qualified by the name which is used to identify the collection.

matching-host

Any specific hosts for which this collection will be used.

Syntax

```
matching-host = <host>
```

Description

Any specific hosts (obtained from the Host header of the HTTP request) for which this collection should be used. If the Host header from a request does not match any specified collection the default collection, as defined by the default-collection-name, will be used.

Shell-style pattern matching characters, including ‘*', ‘?’, ‘\’ and '[]’, can be configured and used in the comparison.

This configuration entry may be repeated multiple times to specify multiple hosts.

Options

<host>
The name of the host for which this collection will be used.

Usage

This stanza entry is optional.

Default value

None

Example

```
matching-host = www.ibm.com
```

server

The name of a Redis server which is contained in this collection.

Syntax

```
server = <server>
```

Description

The name of a Redis server which is contained within this collection. A corresponding '[redis-server:<server>]’ stanza should also be created to define the properties of this server.

This entry may be repeated multiple times, once for each server which is contained within this collection.

Options

<server>
The name associated with the Redis server.

Usage

This stanza entry is required.

Default value

None

Example

```
server = redis-server-a
```

master-authn-server-url

Use this entry to designate the base URL of the master authentication server for this collection of Redis servers.

Syntax

```
master-authn-server-url = <url>
```

Description

The base URL of the master authentication server for this collection of Redis servers. The master authentication server, if specified, will be responsible for the generation of all new sessions for this collection. This configuration entry is designed to be used in an environment where you wish to share sessions across multiple DNS domains. The configuration entry should be of the format: `http{s}://<server>{:<port>}`.

As part of the session sharing flow the URL of the originating server must be dynamically constructed. This URL will be constructed from the first available, and valid, of the following request headers: `Forwarded`, `X-Forwarded-Host` and `X-Forwarded-Proto`, `Host`.

Options

<url>
The base URL of the master authentication server.

Usage

This stanza entry is optional.

Default value

None

Example

```
master-authn-server-url = https://master.ibm.com:9443
```

master-session-code-lifetime

Use this entry to designate the lifetime of a temporary session code.

Syntax

```
master-session-code-lifetime = <lifetime>
```

Description

The maximum number of seconds that a session code, used when communicating the session information from the master authentication server, will remain valid.

Options

<lifetime>
The number of seconds that a temporary session code will remain valid.

Usage

This stanza entry is optional.

Default Value

30

Example

```
master-session-code-lifetime = 30
```

max-pooled-connections

The maximum number of pooled connections to any Redis server.

Syntax

```
max-pooled-collections = <number-of-connections>
```

Description

To help improve performance the established connections to a Redis server can be re-used on subsequent requests. While these connections are not in use they will be cached in a pool of idle connections. This configuration entry defines the maximum number of pooled connections to each Redis server.

Options

<number-of-connections>
An integer value specifying the maximum number of pooled connections for any Redis server.

Usage

This stanza entry is optional.

Default value

50

Example

```
max-pooled-connections = 50
```

connect-timeout

The maximum length of time to wait while establishing a connection to a Redis server.

Syntax

```
connect-timeout = <seconds>
```

Description

The maximum number of seconds to wait for a connection to be established with a server.

Options

<seconds>
The maximum length of time, in seconds, to wait to establish a connection to a Redis server.

Usage

This stanza entry is optional.

Default value

5

Example

```
connect-timeout = 5
```

io-timeout

The maximum length of time to wait for a valid response from a Redis server.

Syntax

```
io-timeout = <seconds>
```

Description

The maximum number of seconds to wait for a valid response from a Redis server.

Options

<seconds>
The maximum length of time, in seconds, to wait for a valid response.

Usage

This stanza entry is optional.

Default value

30

Example

```
io-timeout = 30
```

health-check-interval

The length of time, in seconds, between Redis server health checks.

Syntax

```
health-check-interval = <seconds>
```

Description

The interval, in seconds, between health check requests sent to the Redis server.

Options

<seconds>
The interval, in seconds, between Redis server health checks.

Usage

This stanza entry is optional.

Default value

15

Example

```
health-check-interval = 15
```

[redis-server:<server-name>] stanza

Use the `redis-server:<server-name>` stanza to define the characteristics of a single Redis server. The stanza name is qualified by the name which is used to identify the server.

gsk-attr-name

Syntax

```
gsk-attr-name = {enum | string | number}:id:value
```

Description

Specify additional GSKit attributes to use when initializing an SSL connection to the Redis server. This configuration entry can be specified multiple times. Configure a separate entry for each GSKit attribute.

Options

{enum | string | number}
The GSKit attribute type.

id
The identity associated with the GSKit attribute.

value
The value for the GSKit attribute.

Usage

This stanza entry is optional.

See “Appendix: Supported GSKit attributes” on page 583 for a list of GSKit attributes that can be configured.

You cannot configure the following restricted GSKit attributes:

```
GSK_BASE_CRYPTO_LIBRARY
GSK_KEYRING_FILE
GSK_KEYRING_PW
GSK_KEYRING_STASH_FILE
GSK_KEYRING_LABEL
GSK_ACCELERATOR_NCIPHER_NF
GSK_ACCELERATOR_RAINBOW_CS
GSK_PKCS11_DRIVER_PATH
GSK_PKCS11_TOKEN_LABEL
GSK_PKCS11_TOKEN_PWD
GSK_PKCS11_ACCELERATOR_MODE
GSK_V2_SESSION_TIMEOUT
GSK_V3_SESSION_TIMEOUT
GSK_PROTOCOL_SSLV2
GSK_PROTOCOL_SSLV3
GSK_IO_CALLBACK
GSK_RESET_SESSION_TYPE_CALLBACK
GSK_USE_IO_EVENTS
GSK_USER_DATA
```

If you attempt to modify any of these attributes then an error message will be generated.

Default value

None.

Example

The following entry is for the GSKit attribute GSK_HTTP_PROXY_SERVER_NAME, which has an identity value of 225:

```
gsk-attr-name = string:225:proxy.ibm.com
```

server

The server name or IP address of the Redis server.

Syntax

```
server = <name>
```

Description

The server name or the IP address of the Redis server.

Options

<name>
The server name or the IP address of the Redis server.

Usage

This stanza entry is required.

Default value

None

Example

```
server = redis.ibm.com
```

port

The port on which the Redis server is listening for requests.

Syntax

```
port = <port_number>
```

Description

The port on which the Redis server is listening for requests.

Options

<port_number>
The port which will be used when connecting to the Redis server.

Usage

This stanza entry is optional.

Default value

6379

Example

```
port = 6379
```

password

The password which is used to authenticate to the Redis server.

Syntax

```
password = <password>
```

Description

The password which is used in the '**AUTH**' command when authenticating to the Redis server.

Options

<password>
The password which is used to authenticate to the Redis server.

Usage

This stanza entry is optional.

Default value

None

Example

```
password = passw0rd
```

client-certificate-label

The label of the certificate used to mutually authenticate to the Redis server.

Syntax

```
client-certificate-label = <label>
```

Description

The label associated with the client key which is used to perform mutual authentication with the Redis server. This key must exist in the key file which is used to secure the Redis communication (i.e., it is defined by the ssl-keyfile configuration entry within this stanza).

Options

<label>
The label of the personal certificate to be used for mutual authentication.

Usage

This stanza entry is optional.

Default value

None

Example

```
client-certificate-label = my-cert
```

ssl-keyfile

The name of the key database which contains the certificates and keys used when communicating with the Redis server.

Syntax

```
ssl-keyfile = <keyfile>
```

Description

The name of the key database which is to be used when accessing this server. The key database should contain the CA certificate for the Redis server certificate, and if mutual authentication is in use, any intermediate certificates used to sign the client certificate, and the client key itself.

SSL/TLS will not be used when communicating with the Redis server if no SSL key file is specified.

Options

<keyfile>
The name of the SSL key database.

Usage

This stanza entry is optional.

Default value

None

Example

```
ssl-keyfile = pdsrv.kdb
```

sni-name

The Server Name Indicator (SNI) value used when establishing the secure connection to the Redis server.

Syntax

```
sni-name = <name>
```

Description

The Server Name Indication (SNI) value which is provided when establishing the SSL connection with the Redis server.

Options

<name>
The Server Name Indicator (SNI) value.

Usage

This stanza entry is optional.

Default value

None

Example

```
sni-name = redis.ibm.com
```

username

The name of the user which is used when authenticating to the Redis server.

Syntax

```
username = <user>
```

Description

The name of the user which is used in the '**AUTH**' command when authenticating to the Redis server.

Options

<user>
The name of the user which is used to authenticate to the Redis server.

Usage

This stanza entry is optional.

Default value

None

Example

```
username = user-a
```

[remember-me] stanza

Use the `[remember-me]` stanza to configure the Web Reverse Proxy to remember the username and/or the session.

remember-username-cookie-name

The remember-username-cookie-name configuration entry specifies the name of the cookie which will store the authenticated username.

Syntax

```
remember-username-cookie-name = <cookie_name>
```

Description

The name of the cookie which is used to store the authenticated username. The cookie will be created if the '`remember-username`' field is set to '`true`' in the authentication form. Leave this configuration entry empty if you do not want to enable the ability to remember the username.

Note: The default WebSEAL `login.html` file has sample JavaScript included which will automatically select the '**Remember my username**' checkbox if the `remember-username` cookie has been set. This JavaScript assumes that the cookie name is set to 'verify-access-username'. If a different cookie name is configured the JavaScript should be updated accordingly.

Options

<cookie_name>
The name of the cookie.

Usage

This stanza entry is optional.

Default value

None.

Example

```
remember-username-cookie-name = verify-access-username
```

remember-username-cookie-domain-cookie

Use the remember-username-cookie-domain-cookie stanza entry to enable the username cookie for the domain.

Syntax

```
remember-username-cookie-domain-cookie = {yes|no}
```

Description

Enables the username cookie for the domain.

Options

{yes}
Enables the username cookie for the domain.

{no}
Disables the username cookie for the domain.

Usage

This stanza entry is required

Default value

None

Example

```
remember-username-cookie-domain-cookie = no
```

remember-session-field

The `remember-session-field` configuration entry specifies the name of the field which will store the authenticated session token.

Syntax

```
remember-session-field = [hdr|cookie]:<field_name>
```

Description

The name of the HTTP field which is used to transmit the remember-session token. This field will be added to an authentication response if the 'remember-session' field is set to true in the authentication form, or the 'remember-session' flag is set in the EAI authentication response. Leave this configuration entry empty if you do not want to enable the ability to remember the session.

Options

hdr
Indicates that a HTTP header will be used.

cookie
Indicates that a HTTP cookie will be used.

<field_name>
The name of the HTTP header or cookie.

Usage

This stanza entry is optional.

Default Value

None

Example

```
remember-session-field = cookie:verify-access-persistent-session
```

remember-session-lifetime

Use the `remember-session-lifetime` configuration entry to specify the maximum lifetime, in minutes, for a session token.

Syntax

```
remember-session-lifetime = <number_of_minutes>
```

Description

The number of minutes that the remember-session token will remain valid. A value of -1 indicates that the token will never expire.

Options

<number_of_minutes>
An integer value specifying the number of minutes for which the session token is valid. There is no maximum value. A value of -1 indicates that the token will never expire.

Usage

This stanza entry is required.

Default Value

10080

Example

```
remember-session-lifetime = 10080
```

remember-session-cookie-domain-cookie

Use the `remember-session-cookie-domain-cookie` stanza entry to enable the session cookie for the domain.

Syntax

```
remember-session-cookie-domain-cookie = {yes|no}
```

Description

Enables the session cookie for the domain.

Options

yes
Enables the session cookie for the domain

no
Disables the session cookie for the domain

Usage

This stanza entry is required

Default Value

None

Example

```
remember-session-cookie-domain-cookie = no
```

remember-session-key-label

The `remember-session-key-label` configuration entry specifies the label of the key which will be used to protect the session token.

Syntax

```
remember-session-key-label = <key_label>
```

Description

The key database, which is referenced by the `webseal-cert-keyfile` configuration entry, which will be used to secure the session token.

Options

<key_label>
The label of the key within the `webseal-cert-keyfile` key database to be used.

Usage

This stanza entry is required if a '`remember-session-field`' configuration entry has been specified.

Default Value

None

Example

```
remember-session-key-label = session-key
```

remember-session-attribute-rule

The `remember-session-attribute-rule` configuration entry specifies the rules which are used to determine which credential attributes should be stored in the session token.

Syntax

```
remember-session-attribute-rule =[+|-]<attribute_pattern>
```

Description

The rules which define the credential attributes which will be stored in the remember-session token. This entry may be repeated multiple times, once for each rule which is to be defined.

Each attribute in the credential will be matched against each rule in order until a match is found. The corresponding prefix (+|-) will then be used to control whether the attribute is included or excluded from the remember-session token. If no matching rule is found the attribute will be excluded from the token.

Options

+
Indicates that the attribute should be included.

-
Indicates that the attribute should be excluded.

<attribute_pattern>
The name of the attribute to which this rule applies. The attribute pattern can contain the '*?' pattern matching characters.

Usage

This stanza entry is optional.

Default Value

None

Example

```
remember-session-attribute-rule = +AUTHENTICATION_LEVEL
remember-session-attribute-rule = +AZN_CRED_NETWORK_ADDRESS_STR
```

remember-session-attribute-literal

The `remember-session-attribute-literal` configuration entry specifies the name of any literal values which should be added to the session token.

Syntax

```
remember-session-attribute-literal = <name>:<value>
```

Description

Any literal string attributes which should be added to the session token. This attribute will replace any attributes which might have been added to the token from the credential itself.

This entry may be repeated multiple times, once for each literal attribute which is to be added to the token.

Options

<name>
The name of the attribute to be added to the session token.

<value>
The literal value of the attribute to be added to the session token.

Usage

This stanza entry is optional.

Default Value

None

Example

```
remember-session-attribute-literal = AUTHENTICATION_LEVEL:1
```

[replica-sets] stanza

replica-set

Use the **replica-set** stanza entry to configure WebSEAL to join replica sets that are managed by the distributed session cache server.

Syntax

```
replica-set = replica_set_name
```

Description

If WebSEAL is configured to use the distributed session cache for session storage, the WebSEAL server joins each of the replica sets listed in this stanza. The entries listed here must be replica sets configured on the distributed session cache.

Options

replica_set_name
Replica set name.

Usage

This stanza entry is optional.

Default value

None.

Example

```
replica-set = setA
```

[rsp-header-names] stanza

Defines static HTTP headers are added to every HTTP response from the WebSEAL server.

With this stanza, an administrator can insert some standard security headers into the response, such as **`strict-transport-security`**, **`content-security-policy`**, and **`x-frame-options`**.

Note: The headers that are defined in this stanza will replace any matching headers that might have been added to the response by a junctioned application.

If multiple headers of the same name are specified in this stanza, all but the last of the matching entries will be ignored.

The format of each entry in this stanza is:

```
<header-name> = <header-value>
```

For example:

```
strict-transport-security = max-age=31536000; includeSubDomains
```

A special <*header-value*> of `'%SESSION_EXPIRY%'` can be used to designate a header that will contain the remaining length of time, in seconds, before the current local session expires. This value does not include the overall session timeout for sessions that are managed by the distributed session cache (DSC), but just the length of time before the session expires in the local cache.

For example:

```
session-timeout = %SESSION_EXPIRY%
```

Header names can be added on a per junction basis by specifying the stanza and including the junction ID:

```
[rsp-header-names:/jct_a]
header: value
```

[rsp-header-names:<jct-id>] stanza

Defines static HTTP headers are added to every HTTP response from the WebSEAL server.

With this stanza, an administrator can insert some standard security headers into the response, such as **strict-transport-security**, **content-security-policy**, and **x-frame-options**.

Note: The headers that are defined in this stanza will replace any matching headers that might have been added to the response by a junctioned application.

If multiple headers of the same name are specified in this stanza, all but the last of the matching entries will be ignored.

The format of each entry in this stanza is:

```
<header-name> = <header-value>
```

For example:

```
strict-transport-security = max-age=31536000; includeSubDomains
```

A special <*header-value*> of `'%SESSION_EXPIRY%'` can be used to designate a header that will contain the remaining length of time, in seconds, before the current local session expires. This value does not include the overall session timeout for sessions that are managed by the distributed session cache (DSC), but just the length of time before the session expires in the local cache.

For example:

```
session-timeout = %SESSION_EXPIRY%
```

Header names can be added on a per junction basis by specifying the stanza and including the junction ID:

```
[rsp-header-names:/jct_a]
header: value
```

[rtss-eas] stanza

You can use the **rtss-eas** configuration stanza to configure the EAS that communicates with the RBA server. The runtime security services EAS is used for a particular object if the effective POP for the object has an attribute called **eas-trigger** with an associated value of `trigger_rba_eas`.

Add this stanza and the associated entries if you want to configure runtime security services for Advanced Access Control.

apply-tam-native-policy

Syntax

```
apply-tam-native-policy = {true | false}
```

Description

Determines whether the IBM Security Verify Access for Web ACL policy takes effect.

Options

true
Runtime security services EAS checks with Security Verify Access whether the user has permission to access the resource based on the ACL policy.

false
Runtime security services EAS does not check the Security Verify Access ACL policy to determine whether the user has permission to access the resource.

Usage

This stanza entry is required.

This stanza entry applies to Advanced Access Control.

Default value

None.

Example

```
apply-tam-native-policy = true
```

audit-log-cfg

Syntax

```
audit-log-cfg = <agent>[<parameter>=<value>],[<parameter>=<value>],...
```

Description

Configures audit logging for the runtime security service. You can use the available parameters to configure the logging agents.

Options

<agent>
Specifies the logging agent. The agent controls the logging destination for server events. Valid agents include:

- `stdout`
- `stderr`
- `file`
- `remote`
- `rsyslog`

<parameter>
The different agents support the following configuration parameters:

Table 4. Logging agent configuration parameters

Parameter	Supporting agents
`buffer_size`	`remote`
`compress`	`remote`
dn	`remote`

Table 4. Logging agent configuration parameters (continued)	
Parameter	**Supporting agents**
error_retry	remote, rsyslog
flush_interval	all
hi_water	all
log_id	file, rsyslog
max_event_len	rsyslog
mode	file
path	all
port	remote, rsyslog
queue_size	all
rebind_retry	remote, rsyslog
rollover_size	file
server	remote, rsyslog
ssl_keyfile	rsyslog
ssl_label	rsyslog
ssl_stashfile	rsyslog

Note: For a complete description of the available logging agents and the supported configuration parameters, see the *Security Verify Access: Auditing Guide*.

Usage

This stanza entry is optional.

This stanza entry applies to Advanced Access Control.

Note: You must configure this attribute if you want WebSEAL to log runtime security audit events. If there is no value set, then WebSEAL does not log any audit events for the runtime security service.

Default value

None.

Example

To log audit events in a file called `rtss-audit.log`:

```
audit-log-cfg = file path=/tmp/rtss-audit.log,flush_interval=20,
rollover_size=2000000,queue_size=48
```

To send audit logs to STDOUT:

```
audit-log-cfg = stdout
```

cba-cache-size

Use this entry to specify the maximum number of cached RTSS outcomes.

Syntax

```
cba-cache-size = <max_number_of_entries>
```

Description

Specifies the maximum number of cached RTSS outcomes.

RTSS authorization decisions are only cached if:

- The **CBACacheResult** attribute of the protected object, to which the POP containing the **eas-trigger** attribute has been attached, has a nonzero value.
- The **user-session-ids** attribute in the **[session]** stanza is set to yes.
- The result does not contain an obligation.

The **CBACacheResult** attribute supports the following values:

-1
Cache for the life of the login session.

0
Do not cache (implied default).

>1
Seconds to cache.

When a decision is cached, it will stay cached and be reused even if some of the inputs that affect the decision change on future requests.

A least-recently-used (LRU) algorithm is used to discard older decisions when the cache becomes full.

Options

<max_number_of_entries>
The size of the cache table.

Usage

This stanza entry is optional.

Default value

16384

Example

```
cba-cache-size = 32768
```

cluster-name

Syntax

```
cluster-name = <cluster_name>
```

Description

The name of the runtime security services SOAP cluster that hosts this runtime security SOAP service. You must also specify a corresponding **[rtss-cluster:<cluster>]** stanza, which contains the definition of the cluster.

Options

<cluster_name>
The name of the runtime security services SOAP cluster where the runtime security SOAP service is hosted.

Usage

This stanza entry is required.

This stanza entry applies to Advanced Access Control.

Default value

None.

Example

```
cluster-name = cluster1
```

For this example, there needs to be a corresponding **[rtss-cluster:cluster1]** stanza to define the cluster.

context-id

Specify the context ID that the runtime security services EAS uses when sending XACML requests.

Syntax

```
context-id = {other-policy-id | context-inherited-pop | context-server-name}
```

Description

Specifies the context ID that the runtime security services EAS uses when sending XACML requests to runtime security services. This value must match the service name of the deployed policy.

Note: If the **`context-id`** parameter is not set, it defaults to the WebSEAL server name.

Options

context-inherited-pop
Use the location of the inherited POP for all requests. Use this value if you require multiple policies for different portions of the protected resource tree.

context-server-name
Use the WebSEAL server name for all requests.

other-policy-id
Specify the value for the policy ID for all requests.

Usage

This stanza entry is optional.

This stanza entry applies to Advanced Access Control.

Default value

If there is no value provided for this parameter, it defaults to the WebSEAL server name.

Example

```
context-id = context-inherited-pop
```

provide_700_attribute_ids

Specify whether to enable the attribute IDs from Risk-Based Access version 7.0.

Syntax

```
provide_700_attribute_ids = {true | false}
```

Description

Several attribute IDs for Risk-Based Access were changed from version 7.0. To use the 7.0 attribute IDs, add this entry to your configuration file. Specify this entry under the `[rtss-eas]` stanza.

Options

true
Runtime security services uses the previous attribute IDs from version Risk-Based Access 7.0.

false
Runtime security services uses the new attribute IDs defined for the current version.

Usage

This stanza entry is not required.

This stanza entry applies to Advanced Access Control.

Default value

false

Example

```
provide_700_attribute_ids = true
```

trace-component

Syntax

```
trace-component = <component_name>
```

Description

Specifies the name of the Security Verify Access trace component that the EAS uses.

Options

<component_name>
The name of the Security Verify Access trace component.

Usage

This stanza entry is required.

This stanza entry applies to Advanced Access Control.

Note: The configured component traces the data that passes into the runtime security services EAS, which is governed by the **[azn-decision-info]** stanza. This trace might contain sensitive information.

Default value

None.

Example

```
trace-component = pdweb.rtss
```

use_real_client_ip

Activate the use of the real client IP address for authorization decisions.

Syntax

```
use_real_client_ip = {true | false}
```

Description

Determines whether to activate the current, or real, client IP address. Specify this entry under the `[rtss-eas]` stanza.

If the `client_ip = client_ip` entry exists in the `[azn-decision-info]` stanza, then the current client IP address is activated in the `AZN_CRED_NETWORK_ADDRESS_STR` credential.

In IBM Security Access Manager for Web version 7.0, the value of `AZN_CRED_NETWORK_ADDRESS_STR` contained the client IP address when the user first authenticated and the credential was built. If the IP address changed during the session, the value was not updated. If you want to use this type of client IP address, you have two options:

- Do not add the `client_ip = client_ip` entry to the `[azn-decision-info]` stanza.
- Add the `client_ip = client_ip` entry to the `[azn-decision-info]` stanza. Also, set `use_real_client_ip = false` under the `[rtss-eas]` stanza.

Options

true
Runtime security services EAS uses the current and real IP address.

false
Runtime security services EAS uses the client IP address from when the user first authenticated and the credential was built. If the client IP changes during the session, the client IP is not updated.

Usage

This stanza entry is not required.

This stanza entry applies to Advanced Access Control.

Default value

true

Example

```
use_real_client_ip = false
```

[rtss-cluster:<cluster>] stanza

This stanza contains the configuration entries for the runtime security services SOAP servers.

basic-auth-user

Syntax

```
basic-auth-user = <user_name>
```

Description

Specifies the name of the user for WebSEAL to include in the basic authentication header when communicating with the runtime security services SOAP server.

Options

<user_name>
The user name for WebSEAL to include in the basic authentication header.

Usage

This stanza entry is optional.

Note: Configure this entry if the runtime security services SOAP server is configured to require basic authentication.

Default value

None.

Example

```
basic-auth-user = userA
```

basic-auth-passwd

Syntax

```
basic-auth-passwd = <password>
```

Description

Specifies the password for WebSEAL to include in the basic authentication header when communicating with the runtime security services SOAP server.

Options

<password>
The password that WebSEAL includes in the basic authentication header.

Usage

This stanza entry is optional.

Note: Configure this entry if the runtime security services SOAP server is configured to require basic authentication.

Default value

None.

Example

```
basic-auth-passwd = password
```

handle-idle-timeout

Syntax

```
handle-idle-timeout = <number>
```

Description

Specifies the length of time, in seconds, before an idle handle is removed from the handle pool cache.

Options

<number>
Length of time, in seconds, before an idle handle is removed from the handle pool cache.

Usage

This stanza entry is required.

Default value

None.

Example

```
handle-idle-timeout = 240
```

handle-pool-size

Syntax

```
handle-pool-size = <number>
```

Description

The maximum number of cached handles that WebSEAL uses to communicate with runtime security services SOAP.

Options

<number>
The maximum number of handles that WebSEAL uses for runtime security services SOAP communication.

Usage

This stanza entry is required.

Default value

None.

Example

```
handle-pool-size = 10
```

load-balance

Controls the behavior when multiple servers with the same configured priority are available.

Syntax

```
load-balance = {yes | no}
```

Description

Specifies if round robin load balancing is used when multiple servers are configured with the same priority.

Options

yes
All requests are sent to matching servers in a round-robin fashion.

no
All requests are sent to the first matching server that is available. The matching order is the order they appear in the configuration file.

Usage

This stanza entry is optional.

Default value

The default value is `yes`.

Example

```
load-balance = yes
```

max-wait-time

Use this entry to control the maximum length of time, in seconds, that the request will block while waiting for a server to become available.

Syntax

```
max-wait-time = <number>
```

Description

Specifies the maximum length of time, in seconds, that the request will block while waiting for a server to become available. This configuration entry can be used to help eliminate errors being returned to the client during a server failover.

Options

<number>
Length of time, in seconds, that the request will block while waiting for a server to become available.

Usage

This stanza entry is optional.

Default value

`0`

Example

```
max-wait-time = 0
```

server

Syntax

```
server = {[0-9],}<URL>
```

Description

Specifies a priority level and URL for each runtime security services SOAP server that is a member of this cluster. Multiple server entries can be specified for a given cluster for failover and load balancing.

Options

[0-9]
A digit, 0-9, that represents the priority of the server in the cluster (9 being the highest, 0 being the lowest). If the priority is not specified, a priority of 9 is assumed.

Note: There can be no space between the comma (,) and the URL. If no priority is specified, the comma is omitted.

<URL>
A well-formed HTTP or HTTPS uniform resource locator for the runtime security services (RTSS).

Usage

This stanza entry is required.

Default value

None.

Example

```
server = 9,http://localhost:9080/rtss/authz/services/AuthzService
```

ssl-fips-enabled

Syntax

```
ssl-fips-enabled = {yes|no}
```

Description

Determines whether Federal Information Process Standards (FIPS) mode is enabled with runtime security services SOAP.

Notes:

- If no configuration entry is present, the setting from the global setting, determined by the Verify Access policy server, takes effect.
- The **[rtss-cluster:<*cluster*>] ssl-nist-compliance** setting can override this entry. If **ssl-nist-compliance** is set to yes, FIPS mode processing is automatically enabled.

Options

yes
FIPS mode is enabled.

no
FIPS mode is disabled.

Usage

This stanza entry is required if both of the following conditions are true:

- One or more of the cluster **server** entries use SSL. That is, at least one **server** entry specifies a URL that uses the HTTPS protocol.
- A certificate is required other than the default certificate used by WebSEAL when communicating with the policy server.

Note: If this entry is required, but it is not specified in the **[rtss-cluster:<*cluster*>]** stanza, WebSEAL uses the value in the global **[ssl]** stanza.

Default value

None.

Note: If you want to use a FIPS level that is different to the Verify Access policy server, edit the configuration file and specify a value for this entry.

Example

```
ssl-fips-enabled = yes
```

ssl-keyfile

Syntax

```
ssl-keyfile = <file_name>
```

Description

The name of the key database file that houses the client certificate for WebSEAL to use.

Options

<file_name>
The name of the key database file that houses the client certificate for WebSEAL to use.

Usage

This stanza entry is required if both of the following conditions are true:

- One or more of the cluster **server** entries use SSL. That is, at least one **server** entry specifies a URL that uses the HTTPS protocol.
- A certificate is required other than the default certificate used by WebSEAL when communicating with the policy server.

Default value

None.

Example

```
ssl-keyfile = webseald.kdb
```

ssl-keyfile-label

Syntax

```
ssl-keyfile-label = <label_name>
```

Description

The label of the client certificate in the key database.

Options

<label_name>
Client certificate label name.

Usage

This stanza entry is required if both of the following conditions are true:

- One or more of the cluster **server** entries use SSL. That is, at least one **server** entry specifies a URL that uses the HTTPS protocol.
- A certificate is required other than the default certificate used by WebSEAL when communicating with the policy server.

Default value

None.

Example

```
ssl-keyfile-label = WebSEAL-Test
```

ssl-keyfile-stash

Syntax

```
ssl-keyfile-stash = <file_name>
```

Description

The name of the password stash file for the key database file.

Options

<file_name>
The name of the password stash file for the key database file.

Usage

This stanza entry is required if both of the following conditions are true:

- One or more of the cluster **server** entries use SSL. That is, at least one **server** entry specifies a URL that uses the HTTPS protocol.
- A certificate is required other than the default certificate used by WebSEAL when communicating with the policy server.

Default value

None.

Example

```
ssl-keyfile-stash = sslstashfile.sth
```

ssl-nist-compliance

Use the **ssl-nist-compliance** stanza entry in the **[rtss-cluster:<*cluster*>]** stanza to enable or disable NIST SP800-131A compliance for runtime security services SOAP communication.

Syntax

```
ssl-nist-compliance = {yes|no}
```

Description

Enables or disables NIST SP800-131A compliance for runtime security services SOAP communication.

Enabling NIST SP800-131A compliance results in the following automatic configuration:

- Enables FIPS mode processing.

  Note: When NIST SP800-131A compliance is enabled, FIPS mode processing is enabled regardless of the setting for the **[rtss-cluster:<*cluster*>] ssl-fips-enabled** configuration entry.
- Enables TLS v1.2.

  Note: TLS v1 and TLS v1.1 are not disabled.
- Enables the appropriate signature algorithms.
- Sets the minimum RSA key size to 2048 bytes.

If this **ssl-nist-compliance** configuration entry is not present, WebSEAL uses the global **nist-compliance** setting in the **[ssl]** stanza.

Options

yes
A value of yes enables NIST SP800-131A compliance.

no
A value of no disables NIST SP800-131A compliance.

Usage

This stanza entry is optional.

Default value

no

Example

```
ssl-nist-compliance = no
```

ssl-valid-server-dn

Syntax

```
ssl-valid-server-dn = <DN-value>
```

Description

Specifies the distinguished name of the server (obtained from the server SSL certificate) that WebSEAL can accept.

Options

<DN-value>
The distinguished name of the server (obtained from the server SSL certificate) that WebSEAL accepts. If no value is specified, then WebSEAL considers all domain names valid. You can specify multiple domain names by including multiple **ssl-valid-server-dn** configuration entries.

Usage

This stanza entry is required if both of the following conditions are true:

- One or more of the cluster **server** entries use SSL. That is, at least one **server** entry specifies a URL that uses the HTTPS protocol.
- A certificate is required other than the default certificate used by WebSEAL when communicating with the policy server.

Default value

None.

Example

```
ssl-valid-server-dn = CN=Verify Access,OU=SecureWay,O=Tivoli,C=US
```

timeout

Syntax

```
timeout = <seconds>
```

Description

The length of time (in seconds) to wait for a response from runtime security services SOAP.

Options

<seconds>
The length of time (in seconds) to wait for a response from runtime security services SOAP.

Usage

This stanza entry is required.

Default value

None.

Example

```
timeout = 240
```

[script-filtering] stanza

hostname-junction-cookie

Syntax

```
hostname-junction-cookie = {yes|no}
```

Description

Enables WebSEAL to uniquely identify the cookie used for resolving unfiltered links. This is used when another WebSEAL server has created a junction to this WebSEAL server, using a WebSEAL to WebSEAL junction.

Options

yes
Enable.

no
Disable.

Usage

This stanza entry is optional, but it is included by default in the configuration file.

Default value

no

Example

```
hostname-junction-cookie = no
```

rewrite-absolute-with-absolute

Syntax

```
rewrite-absolute-with-absolute = {yes|no}
```

Description

Enables WebSEAL to rewrite absolute URLs with new absolute URLs that contain the protocol, host, and port (optionally) that represent how the user accessed the WebSEAL server.

Options

yes
Enable.

no
Disable.

Usage

This stanza entry is optional.

Default value

There is no default value, but if the entry is not specified in this configuration file, WebSEAL assumes the value is no.

Example

```
rewrite-absolute-with-absolute = no
```

script-filter

Syntax

```
script-filter = {yes|no}
```

Description

Enables or disables script filtering support. When enabled, WebSEAL can filter absolute URLs encountered in scripts such as JavaScript.

Options

yes
A value of yes means enabled.

no
A value of no means disabled.

Usage

This stanza entry is optional, but is included by default.

Default value

When it is not declared, the value for the **script-filter** functionality is no by default.

Example

```
script-filter = no
```

[server] stanza

allow-shift-jis-chars

Syntax

```
allow-shift-jis-chars = {yes|no}
```

Description

Specifies whether junctions created using -w will allow all Shift-JIS multibyte characters in junction file and path names.

Options

yes
Junctions created using -w will allow all Shift-JIS multibyte characters in junction file and path names.

no
Junction file and path names using Shift-JIS multibyte characters containing the single byte character '\' will be rejected.

Usage

This stanza entry is required.

Default value

no

Example

```
allow-shift-jis-chars = no
```

allow-unauth-ba-supply

Use the **allow-unauth-ba-supply** stanza entry to control whether unauthenticated users can access junctions that were created with the `-b supply` option.

Syntax

```
allow-unauth-ba-supply = {yes|no}
```

Description

This parameter determines access to `-b supply` junctions by unauthenticated users. By default, unauthenticated users are required to log in before they can access any resource on a junctioned server, where that junction was created with the `-b supply` argument.

Options

yes
When **allow-unauth-ba-supply** is set to `yes`, unauthenticated users can access `-b supply` junctions. The basic authentication header that is supplied by WebSEAL in the forwarded request contains the string `unauthenticated` for the value of the header.

no
When **allow-unauth-ba-supply** is set to no, unauthenticated users cannot access `-b supply` junctions. Users receive a login prompt.

Usage

This stanza entry is required.

Default value

no

Example

```
allow-unauth-ba-supply = no
```

allow-unsolicited-logins

Use the **allow-unsolicited-logins** stanza entry to control whether WebSEAL accepts unsolicited login requests.

Syntax

```
allow-unsolicited-logins = {yes | no}
```

Description

This parameter controls whether WebSEAL accepts unsolicited authentication requests. If this parameter is set to no, WebSEAL accepts a login request only if WebSEAL sent the login form to the client to prompt authentication.

Options

yes
When **allow-unsolicited-logins** is set to yes, WebSEAL accepts unsolicited logins.

no
When **allow-unsolicited-logins** is set to no, WebSEAL does not accept unsolicited logins. This setting ensures that WebSEAL always issues a login form to the client as part of the authentication process.

Usage

This stanza entry is optional.

Default value

yes

Example

```
allow-unsolicited-logins = yes
```

auth-challenge-type

Use the **auth-challenge-type** stanza entry to specify a comma-separated list of authentication types that WebSEAL can use to challenge a client for authentication information.

Syntax

```
auth-challenge-type = list
```

Description

Each authentication type can be customized for particular user agent strings. For more information about authentication challenges based on the user agent, see the *IBM Security Verify Access: Web Reverse Proxy Configuration Guide*.

You can customize this configuration item for a particular junction by adding the adjusted configuration item to a **[server:*{jct_id}*]** stanza.

where *{jct-id}* refers to the junction point for a standard junction (including the leading / character) or the virtual host label for a virtual host junction.

Options

list

A comma-separated list of authentication types that is used when challenging a client for authentication information. The supported authentication types include:

- ba
- cert
- eai
- forms
- oidc
- spnego

The corresponding authentication configuration entry (for example, ba-auth) must be enabled for each specified authentication challenge type.

Each authentication type can also be qualified with a set of rules to specify the user agents that receive a given challenge type. These rules are separated by semicolons and placed inside square brackets preceding the authentication type. Each rule consists of a plus (+) or minus (-) symbol to indicate inclusion or exclusion, and the pattern to match on. The pattern can include:

- Alphanumeric characters
- Spaces
- Periods (.)
- Wildcard characters, such as, question mark (?) and asterisk (*)

Usage

This stanza entry is optional.

Default value

By default, the list of authentication challenge types matches the list of configured authentication mechanisms.

```
auth-challenge-type = ba
auth-challenge-type = forms
```

Example

```
auth-challenge-type = ba, forms
auth-challenge-type = [-msie;+ms]ba, [+mozilla*;+*explorer*]forms
```

cache-host-header

Use the **cache-host-header** stanza entry to control whether WebSEAL caches the host and protocol of the original request.

Syntax

```
cache-host-header = {yes|no}
```

Description

By default, when caching an original request, WebSEAL only caches the URL. That is, WebSEAL does not cache the host and protocol of the original request. In this case, when returning a redirect to the original URL, WebSEAL simply redirects to the current host. This causes problems if a request for a protected

resource on one virtual host, hostA, results in an authentication operation being processed on a different virtual host, hostB. In this case, the client is incorrectly redirected to hostB rather than hostA. This behavior can be corrected by enabling this stanza entry so that WebSEAL can cache the host and protocol of the original request to be used for redirection.

Options

yes
WebSEAL caches the host and protocol of the original request in addition to the URL. In this case:

- Both the host and protocol are cached and used in redirects. They cannot be separately managed.
- The protocol is not cached if the host header is not present.
- Requests will only be recovered from the cache if the protocol, the host and the URL *all* match the original request.

Limitations associated with this caching behavior:

- The contents of the existing URL macro will not include the protocol and host. No new macros have been added to represent these elements.
- It is not possible to specify a protocol and host when a switch user administrator specifies a URL.

no
WebSEAL only caches the URL associated with the original request and redirects to the current host.

Usage

This stanza entry is optional.

Default value

no

Example

```
cache-host-header = yes
```

capitalize-content-length

Use the **capitalize-content-length** stanza entry to control whether WebSEAL capitalizes the first letters in the name of the HTTP content-length header.

Syntax

```
capitalize-content-length = {yes|no}
```

Description

This parameter determines whether WebSEAL uses capitalized first letters in the content-length header. That is, whether the name of the HTTP content-length header is `Content-Length` or `content-length`.

NOTE: The Documentum client application expects the name of the HTTP content-length header to be `Content-Length`, with a capitalized "C" and "L".

Options

yes
WebSEAL uses the Documentum-compliant header name `Content-Length`.

no
WebSEAL uses all lower case for the content-length header. That is, `content-length`.

Usage

This stanza entry is optional.

Default value

no

Example

```
capitalize-content-length = yes
```

clear-cookie-jar-on-reauth

Use the **clear-cookie-jar-on-reauth** entry to define whether to clear cookie jar when a new credential is added to a session.

Syntax

```
clear-cookie-jar-on-reauth = {true | false}
```

Description

WebSEAL cookie jar can be configured to manage cookies sent to junctions which are stored in the user session. These cookies should be cleared for all junctions whenever a re-authentication or step-up authentication operation takes place. In addition, when a new Federation Runtime token is sent to a Federation Runtime junction, the cookies for the junction should be cleared. This backwards compatibility configuration item can be used to disable this behavior so that cookie jar is not cleared.

Options

true
Clear the cookie jar when a re-authentication or step-up authentication operation takes place.

false
Do not clear the cookie jar when a re-authentication or step-up authentication operation takes place.

Usage

This stanza entry is optional.

Default value

```
true
```

Example

```
clear-cookie-jar-on-reauth = true
```

client-connect-timeout

Use the **client-connect-timeout** stanza entry to specify the timeout in seconds for a client connection.

Syntax

```
client-connect-timeout = number_of_seconds
```

Description

After the initial connection handshake has occurred, this parameter dictates how long (in seconds) WebSEAL holds the connection open for the initial HTTP or HTTPS request.

Options

number_of_seconds
Must be a positive integer. Other values have unpredictable results and should not be used. Maximum allowed value: 2147483647.

Usage

This stanza entry is required.

Default value

120

Example

```
client-connect-timeout = 120
```

client-ip-rule

The `client-ip-rule` configuration entry specifies the rules which are used to determine whether a client is allowed to connect to the server.

Syntax

```
client-ip-rule =[+|-]<client-ip>
```

Description

The rules which define whether a client is allowed to connect to this server. This entry may be repeated multiple times, once for each rule which is to be defined.

The client IP address of a request will be evaluated against each rule in sequence until a match is found. The corresponding code (+|-) will then be used to determine whether the client connection is accepted. If the client IP matches no configured rules the client connection will be accepted.

Options

+
Indicates that the client is permitted to connect.

-
Indicates that the client is not permitted to connect.

<client-ip>
The IP address of the client. This field can contain the '*?' pattern matching characters.

Usage

This stanza entry is optional.

Default Value

None

Example

```
client-ip-rule = +10.10.10.*
client-ip-rule = -*
```

chunk-responses

Syntax

```
chunk-responses = {yes|no}
```

Description

Enables WebSEAL to write chunked data to HTTP/1.1 clients. his can improve performance by allowing connections to be reused even when the exact response length is not known before the response is written.

Options

yes
Enable.

no
Disable.

Usage

This stanza entry is required.

Default value

yes

Example

```
chunk-responses = yes
```

concurrent-session-threads-hard-limit

Use the **concurrent-session-threads-hard-limit** stanza entry to specify the maximum number of concurrent threads that a single user session can use before WebSEAL stops processing requests for the session.

Syntax

```
concurrent-session-threads-hard-limit = number_of_threads
```

Description

The maximum number of concurrent threads that a single user session can consume. When a user session reaches its thread limit, WebSEAL stops processing any new requests for the user session and returns an error to the client.

If you do not specify a value for this entry, there is no limit to the number of concurrent threads that a user session can consume.

Options

number_of_threads

The maximum number of concurrent threads that a single user session can consume before WebSEAL returns an error.

Usage

This stanza entry is optional.

Default value

Unlimited.

Example

```
concurrent-session-threads-hard-limit = 10
```

concurrent-session-threads-soft-limit

Use the **concurrent-session-threads-soft-limit** stanza entry to specify the maximum number of concurrent threads that a single user session can use before WebSEAL generates warning messages.

Syntax

```
concurrent-session-threads-soft-limit  = number_of_threads
```

Description

The maximum number of concurrent threads that a single user session can consume before WebSEAL generates warning messages. WebSEAL continues processing requests for this session until it reaches the configured **concurrent-session-threads-hard-limit** (also in the **[server]** stanza).

Options

number_of_threads

Integer value representing the maximum number of concurrent threads that a single session can consume before WebSEAL generates warning messages.

Usage

This stanza entry is optional.

Default value

Unlimited.

Example

```
concurrent-session-threads-soft-limit  = 5
```

connection-request-limit

Use the **connection-request-limit** stanza entry to specify the maximum number of requests that WebSEAL processes on a single persistent connection.

Syntax

```
connection-request-limit = number_of_requests
```

Description

Specifies the maximum number of requests that will be processed on a single persistent connection.

Options

number_of_requests

The maximum number of requests that will be processed on a single persistent connection.

Usage

This stanza entry is required.

Default value

100

Example

```
connection-request-limit = 100
```

cope-with-pipelined-request

Use the **cope-with-pipelined-request** stanza entry to control whether WebSEAL closes the connection and notifies the browser if it detects pipelined requests.

Syntax

```
cope-with-pipelined-request = {yes|no}
```

Description

WebSEAL does not support pipelined requests from browsers. If this option is set to yes, when WebSEAL detects pipelined requests it will close the connection and inform the browser that is should re-send the pipelined requests in a normal manner. This parameter should always be set to yes unless the previous WebSEAL behavior is required.

Options

yes
Enable.

no
Disable.

Usage

This stanza entry is required.

Default value

yes

Example

```
cope-with-pipelined-request = yes
```

decode-query

Use the **decode-query** stanza entry to validate the query string in requests according to the **utf8-qstring-support-enabled** entry.

Syntax

```
decode-query = {yes|no}
```

Description

Validates the query string in requests according to the **utf8-qstring-support-enabled** parameter.

Options

yes
When **decode-query** is set to yes WebSEAL validates the query string in requests according to the **utf8-qstring-support-enabled** parameter. Otherwise, WebSEAL does not validate the query string.

no
When **decode-query** is set to no, then dynurl must be disabled. To disable dynurl, see “dynurl-map” on page 412.

Usage

This stanza entry is required.

Default value

yes

Example

```
decode-query = yes
```

disable-advanced-filtering

Use the **disable-advanced-filtering** stanza entry to control whether encoded URLs are filtered.

Syntax

```
disable-advanced-filtering = {true | false}
```

Description

Enable or disable advanced filtering, which filters encoded URLs.

Options

true
Advanced filtering is disabled.

false
Advanced filtering is enabled.

Usage

This stanza entry is optional.

Default value

`false`

Example

```
disable-advanced-filtering = false
```

disable-timeout-reduction

Use the **disable-timeout-reduction** stanza entry to control whether WebSEAL reduces the timeout duration of threads to help control the number of active worker threads.

Syntax

```
disable-timeout-reduction = {yes|no}
```

Description

By default, WebSEAL automatically reduces the timeout duration for threads as the number of in-use worker threads increases. The timeout duration is the maximum length of time that a persistent connection with the client can remain inactive before WebSEAL terminates the connection.

This configuration option determines whether WebSEAL reduces the timeout duration to help control the number of active worker threads. This option is available on all platforms.

Options

yes
Disables the timeout reduction done by WebSEAL as the number of worker threads in-use increases.

no
WebSEAL performs timeout reduction as the number of worker threads in-use increases.

Usage

This stanza entry is optional.

Default value

no

Example

```
disable-timeout-reduction = yes
```

disable-timeout-reduction

Use the **disable-timeout-reduction** stanza entry to control whether WebSEAL reduces the timeout duration of threads to help control the number of active worker threads.

Syntax

```
disable-timeout-reduction = {yes|no}
```

Description

By default, WebSEAL automatically reduces the timeout duration for threads as the number of in-use worker threads increases. The timeout duration is the maximum length of time that a persistent connection with the client can remain inactive before WebSEAL terminates the connection.

This configuration option determines whether WebSEAL reduces the timeout duration to help control the number of active worker threads. This option is available on all platforms.

Options

yes
Disables the timeout reduction done by WebSEAL as the number of worker threads in-use increases.

no
WebSEAL performs timeout reduction as the number of worker threads in-use increases.

Usage

This stanza entry is optional.

Default value

no

Example

```
disable-timeout-reduction = yes
```

double-byte-encoding

Use the **double-byte-encoding** stanza entry to control whether WebSEAL treats all encoded characters in URLs as Unicode encoded.

Syntax

```
double-byte-encoding = {yes|no}
```

Description

Specifies whether WebSEAL assumes that encoded characters within URLs are always encoded in Unicode, and do not contain UTF-8 characters.

Options

yes
WebSEAL assumes that encoded characters within URLs are always encoded in Unicode, and do not contain UTF-8 characters.

no
WebSEAL does not assume that encoded characters within URLs are always encoded in Unicode, and do not contain UTF-8 characters.

Usage

This stanza entry is required.

Default value

no

Example

```
double-byte-encoding = no
```

dynurl-allow-large-posts

Use the **dynurl-allow-large-posts** stanza entry to control whether WebSEAL accepts POST requests with a body larger than the number of bytes specified by the **request-body-max-read** entry.

Syntax

```
dynurl-allow-large-posts = {yes|no}
```

Description

Allows or disallows POST requests larger than the current value for the stanza entry **request-body-max-read** in the **[server]** stanza.

Options

yes
When set to `yes`, WebSEAL compares only up to **request-body-max-read** bytes of POST request to the URL mappings contained in dynurl configuration file (`dynurl.conf`).

no
When set to no, WebSEAL disallows POST requests with a body larger than **request-body-max-read**.

Usage

This stanza entry is required.

Default value

no

Example

```
dynurl-allow-large-posts = no
```

dynurl-map

Use the **dynurl-map** stanza entry to specify the file that contains mappings for URLs to protected objects.

Syntax

```
dynurl-map = file_name
```

Description

Specifies the file that contains mappings for URLs to protected objects. To disable the `dynurl-map`, leave the value blank.

Options

file_name
The name of the file that contains mappings for URLs to protected objects.

Usage

This stanza entry is optional.

Default value

None, but this entry is usually configured to `dynurl.conf`.

Example

```
dynurl-map = dynurl.conf
```

enable-http2

Use the **enable-http2** stanza entry to enable HTTP/2 support.

Syntax

```
enable-http2 = {yes|no}
```

Description

Set this stanza entry to `yes` to enable HTTP/2 encoded connections from browsers. This setting only affects the "default" interface defined in this stanza.

Note: The cipher suite must also be set to TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 or higher to support HTTP/2 client connections.

Options

yes
Enable HTTP/2 support.

no
Disable HTTP/2 support.

Usage

This stanza entry is optional.

Default value

no

Example

```
enable-http2 = yes
```

enable-IE6-2GB-downloads

Use the **enable-IE6-2GB-downloads** stanza entry to disable the **HTTP Keep-Alives Enabled** option for responses that are sent back to Internet Explorer, version 6, client browsers. This configuration allows clients that are using Microsoft Internet Explorer, version 6.0 to download files greater than 2 GB, but less than 4 GB.

Syntax

```
enable-IE6-2GB-downloads = {yes|no}
```

Description

Allows you to disable the **HTTP Keep-Alives Enabled** option for responses sent back to Internet Explorer, version 6, client browsers. The primary purpose of this is to allow WebSEAL to mimic the Internet Information Services workaround published at http://support.microsoft.com/kb/298618. This will allow clients using Microsoft Internet Explorer, version 6.0, to download files greater than 2GB, but less than 4GB.

NOTE:

- This stanza entry is not necessary for Internet Explorer 7 or for other non-Microsoft browsers.
- Enabling this workaround will cause WebSEAL to not use persistent connections for Internet Explorer, version 6, client connections when the data to be returned in the response is >= 2GB in length.

Options

yes
Disables the `HTTP Keep-Alives Enabled` option, allowing clients using Internet Explorer, version 6, to download files greater than 2GB, but less than 4GB.

no
The **HTTP Keep-Alives Enabled** is not disabled.

Usage

This stanza entry is optional.

Default value

no

Example

```
enable-IE6-2GB-downloads = yes
```

filter-nonhtml-as-xhtml

Use the **filter-nonhtml-as-xhtml** stanza entry to enable tag-based filtering of static URLs for new MIME types added to the **[filter-content-types]** stanza.

Syntax

```
filter-nonhtml-as-xhtml = {yes|no}
```

Description

Enable tag-based filtering of static URLs for new MIME types added to the **[filter-content-types]** stanza.

Options

yes
Enable tag-based filtering of static URLs for new MIME types added to the **[filter-content-types]** stanza

no
Disable tag-based filtering of static URLs for new MIME types added to the **[filter-content-types]** stanza

Usage

This stanza entry is required.

Default value

no

Example

```
filter-nonhtml-as-xhtml = no
```

follow-redirects-for

WebSEAL can examine 302 responses and process the redirects internally if they are destined for the current server. This configuration entry controls the requests for which this redirect function is enabled.

Syntax

```
follow-redirects-for = { pattern | !LRR! | !LOCATION! <pattern> }
```

Description

The configuration entry is used to determine whether a 302 redirect should be handled internally or sent back to the client. Multiple patterns can be specified by including multiple configuration entries of the same name.

Options

pattern
The requests for which the WebSEAL internal redirect function is enabled.

You can use shell-style pattern matching characters "*", "?", "\", and "[]" in the pattern.

!LRR!
Match any request that results in a Local Response Redirect action.

!LOCATION! <pattern>
Match the returned `location` HTTP header. You can use shell-style pattern matching characters "*", "?", "\", and "[]" in the pattern.

Usage

This entry is required if the **maximum-followed-redirects** entry in the **[server]** stanza is set to a value other than `0`.

Default value

None.

Example

```
follow-redirects-for = GET /jct/cgi-bin/eai*
follow-redirects-for = !LRR!
follow-redirects-for = !LOCATION! /jct/cgi-bin/eai*
```

force-tag-value-prefix

Use the **force-tag-value-prefix** stanza entry to control whether WebSEAL prefixes each HTTP-Tag-Value attribute that is set on the junction object with "tagvalue_" before it is added to the credential.

Syntax

```
force-tag-value-prefix = {yes|no}
```

Description

Determines whether each attribute name set in a junction object's HTTP-Tag-Value is automatically prefixed with "tagvalue_" before it is placed in the credential. This prohibits access to credential attributes that do not have names beginning with "tagvalue_" such as AUTHENTICATION_LEVEL. When this options

set to *no*, the automatic prefixing of "tagvalue_" will not occur so that all credential attributes can be specified in HTTP-Tag-Value.

Options

yes
Enable the automatic prefixing of "tagvalue_" to each attribute name set in a junction object's HTTP-Tag-Value.

no
Disable the automatic prefixing of "tagvalue_" so that all credential attributes can be specified in HTTP-Tag-Value.

Usage

This stanza entry is required.

Default value

yes

Example

```
force-tag-value-prefix = yes
```

http

Syntax

```
http = {yes|no}
```

Description

Specifies whether HTTP requests will be accepted by the WebSEAL server. This value is set by the administrator during WebSEAL server configuration.

Options

yes
Accept HTTP requests.

no
Do not accept HTTP requests.

Usage

This stanza entry is required.

Default value

no

Example

```
http = yes
```

http2-max-connections

Use the **http2-max-connections** stanza entry to define the max number of connections allowed when HTTP/2 support is enabled.

Syntax

```
http2-max-connections = number_of_connections
```

Description

This stanza entry controls the maximum number of network connections from HTTP/2 enabled browsers. When the max connections is reached, additional connections will be closed without any I/O. This is per interface/port (HTTP and HTTPS). So if both HTTP and HTTPS are enabled, then the total max connections would be double this. This setting only affects the "default" interface defined in this stanza.

Options

number_of_connections
The max number of connections.

Usage

This stanza entry is optional.

Default value

None.

Example

```
http2-max-connections=100
```

http2-max-concurrent-streams

Use the **http2-max-concurrent-streams** stanza entry to set the maximum number of simultaneous multiplexed streams WebSEAL will accept per HTTP/2 network connection.

Syntax

```
http2-max-concurrent-streams = number_of_streams
```

Description

This stanza entry sets the maximum number of simultaneous multiplexed streams WebSEAL will accept per HTTP/2 network connection to a junctioned server.

Note:

- Each stream will have a **http2-initial-window-size** byte buffer.
- Each stream will need a worker-thread to process the one request or response that is sent over it before it is ended.

This configuration item may be customized for a particular junction by adding the adjusted configuration item to a `[junction:{jct_id}]` stanza, where `'{jct-id}'` refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

Options

number_of_steams
The maximum number of simultaneous multiplexed streams WebSEAL will accept per HTTP/2 network connection. A value of -1 denotes the unlimited setting and is not recommended in a production WebSEAL environment as memory use in WebSEAL would be unbounded.

Usage

This stanza entry is optional.

Default value

None.

Example

```
http2-max-concurrent-streams = 100
```

http2-max-connection-duration

Use the **http2-max-connection-duration** stanza entry to set the maximum duration in seconds for an HTTP/2 connection.

Syntax

```
http2-max-connection-duration = number_of_seconds
```

Description

The connection will be closed if this limit is reached. This setting applies to HTTP/2 connections for all interfaces.

Options

number_of_seconds
The max number of seconds an HTTP/2 connection can stay open.

Usage

This stanza entry is optional.

Default value

None.

Example

```
http2-max-connection-duration = 120
```

http2-header-table-size

Use the **http2-header-table-size** stanza entry to define the max header table size for an HTTP/2 network connection.

Syntax

```
http2-header-table-size = table_size
```

Description

This stanza entry defines the maximum size in bytes that WebSEAL accepts for header compression table (RFC 7541). There is one table per HTTP/2 network connection to a junctioned server.

This configuration item may be customized for a particular junction by adding the adjusted configuration item to a `[junction:{jct_id}]` stanza, where `'{jct-id}'` refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

Options

table_size
The maximum size in bytes that WebSEAL will accept for header compression table.

Usage

This stanza entry is optional.

Default value

None.

Example

```
http2-header-table-size = 4096
```

http2-max-header-list-size

Use the **http2-max-header-list-size** stanza entry to define the maximum size of headers that can be sent in a request on an HTTP/2 stream.

Syntax

```
http2-max-header-list-size = number_of_bytes
```

Description

This stanza entry defines the maximum size of headers in bytes that can be sent in a request on an HTTP/2 stream to a junctioned server. A value of -1 denotes the unlimited setting and is not recommended in a production WebSEAL environment as memory use in WebSEAL would be unbounded. If this entry is not set, it will default to the value of **[server] max-client-read**.

This configuration item may be customized for a particular junction by adding the adjusted configuration item to a `[junction:{jct_id}]` stanza, where `'{jct-id}'` refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

Options

number_of_bytes
The maximum size of headers that can be sent in a request on an HTTP/2 stream.

Usage

This stanza entry is optional.

Default value

The value of the **max-client-read** entry in the **[server]** stanza.

Example

```
http2-max-header-list-size = 32768
```

http2-idle-timeout

Use the **http2-idle-timeout** stanza entry to set the amount of time an HTTP/2 connection can be idle (not processing any requests).

Syntax

```
http2-idle-timeout = number_of_seconds
```

Description

The connection will be closed if it is idle for this amount of time. This setting applies to HTTP/2 connections for all interfaces.

Options

number_of_seconds
The max number of seconds an HTTP/2 connection can be idle.

Usage

This stanza entry is optional.

Default value

None.

Example

```
http2-idle-timeout=20
```

http2-initial-window-size

Use the **http2-initial-window-size** stanza entry to define the maximum number of unacknowledged bytes WebSEAL can accept per active multiplexed stream.

Syntax

```
http2-initial-window-size = number_of_bytes
```

Description

This stanza entry defines the maximum number of unacknowledged bytes WebSEAL can accept per active multiplexed stream to a junctioned server.

This configuration item may be customized for a particular junction by adding the adjusted configuration item to a `[junction:{jct_id}]` stanza, where `'{jct-id}'` refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

Options

number_of_bytes
The maximum number of unacknowledged bytes WebSEAL can accept per active multiplexed stream.

Usage

This stanza entry is optional.

Default value

None.

Example

```
http2-initial-window-size = 65535
```

http2-max-frame-size

Use the **`http2-max-frame-size`** stanza entry to define the maximum size of the body of a single HTTP/2 protocol frame sent over the HTTP/2 network connection.

Syntax

```
http2-max-frame-size = number_of_bytes
```

Description

This stanza entry defines the maximum size of the body of a single HTTP/2 protocol frame sent over the HTTP/2 network connection to a junctioned server.

This configuration item may be customized for a particular junction by adding the adjusted configuration item to a `[junction:{jct_id}]` stanza, where `'{jct-id}'` refers to the junction point for a standard junction (include the leading '/'), or the virtual host label for a virtual host junction.

Options

number_of_bytes
The maximum size of the body of a single HTTP/2 protocol frame that can be sent over the HTTP/2 network connection.

Usage

This stanza entry is optional.

Default value

None.

Example

```
http2-max-frame-size = 16384
```

http-method-disabled-local

Syntax

```
http-method-disabled-local = [HTTP_methods]
```

Description

Specifies the HTTP methods that WebSEAL blocks when processing HTTP requests for local resources. By default, WebSEAL blocks the TRACE HTTP method.

Options

HTTP_methods
A comma-separated list of HTTP methods that are blocked when requesting local resources.

Usage

This stanza entry is required.

Default value

TRACE

Example

```
http-method-disabled-local = TRACE
```

http-method-disabled-remote

Syntax

```
http-method-disabled-remote = [HTTP_methods]
```

Description

Specifies the HTTP methods that WebSEAL blocks when processing HTTP requests for junctioned resources. By default, WebSEAL blocks the TRACE HTTP method.

Options

HTTP_methods
A comma-separated list of HTTP methods that are blocked when requesting remote resources.

Usage

This stanza entry is required.

Default value

TRACE

Example

```
http-method-disabled-remote = TRACE
```

http-port

Syntax

```
http-port = port_number
```

Description

Port on which WebSEAL listens for HTTPS requests. This value is set during WebSEAL configuration. When the default HTTP port is already in use, WebSEAL configuration suggests the next available (unused) port number.

Options

port_number
The administrator can modify this number. Valid values include any port number not already in use on the host.

Usage

This stanza entry is required.

Default value

80

Example

```
http-port = 80
```

http-proxy-protocol

Use the `http-proxy-protocol` stanza entry to enable proxy protocol support for HTTP requests.

Syntax

```
http-proxy-protocol = {true|false}
```

Description

Set this stanza entry to `true` to enable proxy protocol support from clients for HTTP requests. This setting only affects the "default" interface defined in this stanza.

Options

`true`
Enable proxy protocol support.

`false`
Disable proxy protocol support.

Default Value

`false`

Example

```
http-proxy-protocol = true
```

https

Syntax

```
https = {yes|no}
```

Description

Specifies whether HTTPS requests will be accepted by the WebSEAL server. This value is set by the administrator during WebSEAL server configuration.

Options

yes
Accept HTTPS requests.

no
Do not accept HTTPS requests.

Usage

This stanza entry is required.

Default value

no

Example

```
https = yes
```

https-port

Syntax

```
https-port = port_number
```

Description

Port on which WebSEAL listens for HTTPS requests. This value is set during WebSEAL configuration. When the default port is already in use, WebSEAL configuration suggests the next available (unused) port number.

Options

port_number
The administrator can modify this number. Valid values include any port number not already in use on the host.

Usage

This stanza entry is required.

Default value

443

Example

```
https-port = 443
```

https-proxy-protocol

Use the `https-proxy-protocol` stanza entry to enable proxy protocol support for HTTPS requests.

Syntax

```
https-proxy-protocol = {true|false}
```

Description

Set this stanza entry to `true` to enable proxy protocol support from clients for HTTPS requests. This setting only affects the "default" interface defined in this stanza.

Options

`true`
Enable proxy protocol support.

`false`
Disable proxy protocol support.

Default Value

`false`

Example

```
https-proxy-protocol = true
```

ignore-missing-last-chunk

Syntax

```
ignore-missing-last-chunk = {yes|no}
```

Description

Controls whether WebSEAL ignores a missing last chunk in a data-stream from a backend server that is using chunked transfer-encoding.

Options

yes
WebSEAL will ignore a missing last-chunk in a data-stream from a backend server that is using chunked transfer-encoding. This matches the behavior in prior releases of WebSEAL.

no
WebSEAL will RST (reset) the connection to the front-end browser if the last-chunk is not present.

Usage

This stanza entry is optional.

Default value

no

Example

```
ignore-missing-last-chunk = yes
```

intra-connection-timeout

Syntax

```
intra-connection-timeout = number_of_seconds
```

Description

This parameter affects request and response data sent as two or more fragments. The parameter specifies the timeout (in seconds) between each request data fragment after the first data fragment is received by WebSEAL. The parameter also governs the timeout between response data fragments after the first data fragment is returned by WebSEAL.

Options

number_of_seconds
If the value of this parameter is set to 0 (or not set), connection timeouts between data fragments are governed instead by the **client-connect-timeout** parameter. The exception to this rule occurs for responses returned over HTTP (TCP). In this case, there is no timeout between response fragments. If a connection timeout occurs on a non-first data fragment due to the **intra-connection-timeout** setting, a TCP RST (reset) packet is sent.

Usage

This stanza entry is required.

Default value

60

Example

```
intra-connection-timeout = 60
```

io-buffer-size

Syntax

```
io-buffer-size = number_of_bytes
```

Description

Positive integer value that indicates the buffer size, in bytes, for low-level reads from and writes to a client.

Options

number_of_bytes

Positive integer value that indicates the buffer size, in bytes, for low-level reads from and writes to a client.

The minimum value is 1. WebSEAL does not impose a maximum value.

A small value (for instance, 10 bytes) can hurt performance by causing frequent calls to the low-level read/write APIs. Up to a certain point, larger values improve performance because they correspondingly reduce the calls to the low-level I/O functions.

However, the low-level I/O functions might have their own internal buffers, such as the TCP send and receive buffers. When **io-buffer-size** exceeds the size of those buffers, there is no longer any performance improvement because those functions read only part of the buffer at the time.

Reasonable values for **io-buffer-size** range from 1 - 16 kB. Values smaller than this range causes calling the low-level I/O functions too frequently. Values larger than this range wastes memory. A 2 MB I/O buffer size uses 4 MB for each worker thread that communicates with the client, since there is an input and output buffer.

Usage

This stanza entry is required.

Default value

4096

Example

```
io-buffer-size = 4096
```

ip-support-level

Syntax

```
ip-support-level = {displaced-only|generic-only|displaced-and-generic}
```

Description

Controls the amount of network information stored in a credential by specifying the required IP level.

Options

displaced-only
WebSEAL only generates the IPv4 attribute when building user credentials and when authenticating users through external authentication.

generic-only
WebSEAL only generates new generic attributes that support both IPv4 and IPv6 when building user credentials and when authenticating users through external authentication.

displaced-and-generic
Both sets of attribute types (produced by displaced-only and generic-only) are used when building user credentials and when authenticating users through external authentication.

Usage

This stanza entry is required.

Default value

`generic-only`

Example

```
ip-support-level = generic-only
```

ipv6-support

Syntax

```
ipv6-support = {yes|no}
```

Description

Enable/disable WebSEAL support for IPv6 format.

Options

yes
Enable WebSEAL support for IPv6 format.

no
Disable WebSEAL support for IPv6 format.

Usage

This stanza entry is required.

Default value

`yes`

Example

```
ipv6-support = yes
```

late-lockout-notification

Syntax

```
late-lockout-notification = {yes|no}
```

Description

WebSEAL returns a server response error page (`acct_locked.html`) that notifies the user of the penalty for reaching or exceeding the maximum value set by the **max-login-failures** policy. This stanza entry specifies whether this notification occurs when the user reaches the **max-login-failures** limit, or at the next login attempt after reaching the limit.

Options

yes
: Upon reaching the maximum value set by the **max-login-failures** policy, WebSEAL returns another login prompt to the user. WebSEAL does not send the account disabled error page to the user until the next login attempt. This response represents pre-version 6.0 behavior for the **max-login-failures** policy.

no
: Upon reaching the maximum value set by the **max-login-failures** policy, WebSEAL immediately sends the account disabled error page to the user.

Usage

Required

Default value

The default for new installations is `no`. The default for migrated installations is `yes`.

Example

```
late-lockout-notification = yes
```

max-client-read

Syntax

```
max-client-read = number_of_bytes
```

Description

Specifies the maximum number of bytes of request line and header information that WebSEAL holds in internal buffers when reading an HTTP request from a client. One purpose for max-client-read is to help protect WebSEAL from denial-of-service attacks.

As of Security Verify Access WebSEAL 6.0, the max-client-read stanza entry no longer impacts the request-body-max-read and request-max-cache stanza entries.

Options

number_of_bytes

The minimum value for this parameter is 32678 bytes. If the total size of the request line and headers is greater than the value specified for this parameter, WebSEAL closes the connection without reading any more data or sending any response to the client.

If the value is set to a number below 32768, the value is ignored and a value of 32768 is used. There is no maximum value. URL and header information in a typical request rarely exceeds 2048 bytes.

Usage

This stanza entry is required.

Default value

32768

Example

```
max-client-read = 32768
```

max-file-cat-command-length

Syntax

```
max-file-cat-command-length = number_of_bytes
```

Description

Specifies the maximum size of the file, specified in bytes, which may be returned from the **file cat** server task command.

If the value of this parameter is less than the size of the file specified in the **file cat** command, the returned file will be truncated. This parameter takes precedence over the optional **-max** *bytes* value in the **file cat** command.

Options

number_of_bytes

The maximum size of the file, specified in bytes, which may be returned from the **file cat** command.

Usage

This stanza entry is required.

Default value

1024

Example

```
max-file-cat-command-length = 512
```

maximum-followed-redirects

Sets the maximum number of consecutive 302 redirects that are followed internally before WebSEAL concedes and passes the response back to the client.

Syntax

```
maximum-followed-redirects = number
```

Description

In some situations, WebSEAL can send a 302 redirect to the client, which causes a new request to be sent to the same WebSEAL server. This situation can occur for WebSEAL generated responses (for example, the redirect after a forms-based authentication) or for responses that come from junctioned Web servers.

WebSEAL can recognize the 302 redirects that are destined for the same server and handle these redirects internally. Enabling this function can help improve the performance of mobile applications by reducing network communication between WebSEAL and the client.

Use the **maximum-followed-redirects** entry to set the maximum number of redirects that WebSEAL handles internally before it sends the redirects to the client for processing.

Options

number
The maximum number of consecutive 302 redirects that are followed internally before WebSEAL concedes and passes the response back to the client.

Note: A value of 0 indicates that all 302 redirects are sent back to the client for processing.

Usage

This entry is optional. If this entry is set to a value other than 0, you must also set the **redirect-methods** entry in the **[server]** stanza.

Default value

0

Example

```
maximum-followed-redirects = 0
```

max-idle-persistent-connections

Syntax

```
max-idle-persistent-connections = number_of_connections
```

Description

The maximum number of idle client persistent connections. Use a value less than the maximum number of connections supported by WebSEAL to ensure that the idle connections do not consume all the available connections.

Options

number_of_connections
Integer value indicating the maximum number of idle client persistent connections.

Usage

This stanza entry is required.

Default value

512

Example

```
max-idle-persistent-connections = 512
```

max-idle-persistent-connections-threshold

Use this entry to set the warning threshold for the number of idle client persistent connections.

Syntax

```
max-idle-persistent-connections-threshold = number_of_connections
```

Description

When the number of idle client persistent connections reaches this threshold, a warning message will be sent to the WebSEAL log file. A value of `-1` means that no warning will be displayed.

Options

number_of_connections
Integer value indicating the warning threshold for the number of idle client persistent connections.

Usage

This stanza entry is required.

Default value

`-1`

Example

```
max-idle-persistent-connections-threshold = -1
```

max-ratelimiting-buckets

Syntax

```
max-ratelimiting-buckets = maximum number of requests made for a client and policy stored in
memory cache
```

Description

max-ratelimiting-buckets applies only to the in-memory cache. WebSEAL does not enforce a limit on the number of entries created on the Redis server.

This information is stored in a cache that has a size limit. When this limit is exceeded the oldest entry is ejected.

Options

Number rate limiting policy

A base rate limiting policy on all URLs, methods, and IPs. It can be used as a matching criteria to ensure that a user is rate-limited before they can saturate the cache.

Usage

This stanza entry is optional.

Default value

`0`

Example

```
max-ratelimiting-buckets = 16384
```

max-shutdown-quiesce-wait-time

The `max-shutdown-quiesce-wait-time` configuration entry controls the length of time, on shutdown, that the server will wait for active Web requests to be finalized.

Syntax

```
max-shutdown-quiesce-wait-time = seconds_to_wait
```

Description

Use this configuration to define the maximum number of seconds to wait, upon shutdown, for outstanding Web requests to complete processing before the server is terminated. The server will be terminated if there are no outstanding requests to be processed, or if it has waited the configured maximum number of seconds - whichever comes first. This gives the server a chance to finalize active requests before shutting down.

Options

`seconds_to_wait`
The number of seconds to wait, on shutdown, for outstanding Web requests to be finalized.

Usage

This stanza entry is optional.

Default Value

`10`

Example

```
max-shutdown-quiesce-wait-time = 10
```

network-interface

Syntax

```
network-interface = ip-address
```

Description

Specify an alternative IP address to be used by this instance of WebSEAL. This allows two or more WebSEAL instances to use different IP addresses and host names when running on the same machine .

Options

ip-address
IP address.

Usage

This stanza entry is optional.

Default value

`0.0.0.0`

Example

```
network-interface = 9.0.0.9
```

persistent-con-timeout

Syntax

```
persistent-con-timeout = number_of_seconds
```

Description

HTTP/1.1 connection timeout, in seconds. This setting affects connections to clients, not to backend server systems.

Options

number_of_seconds
HTTP/1.1 connection timeout, in seconds. Must be a positive integer. Other values have unpredictable results and should not be used. Maximum allowed value: 2147483647.

A value of `0` causes WebSEAL to set the **'Connection: close'** header and then close the connection on every response. If the value of this stanza entry is set to `0`, the connection does not remain open for future requests.

Usage

This stanza entry is required.

Default value

5

Example

```
persistent-con-timeout = 5
```

preserve-base-href

Syntax

```
preserve-base-href = {yes|no}
```

Description

Specifies whether WebSEAL will remove all BASE HREF tags from filtered HTML documents and prepend the base tag to filtered links.

Options

yes
When set to yes, WebSEAL filters the BASE HREF tag.

no
When set to no, WebSEAL removes BASE HREF tags.

Usage

This stanza entry is required.

Default value

no

Example

```
preserve-base-href = no
```

preserve-base-href2

Syntax

```
preserve-base-href2 = {yes|no}
```

Description

Used in conjunction with the **preserve-base-href** option to specify the level of filtering on the BASE HREF tags.

NOTE: This option has no effect unless **preserve-base-href** (also in the **[server]** stanza) is set to yes.

Options

yes
When set to yes, WebSEAL only performs the minimum filtering of the BASE HREF tag necessary to insert the WebSEAL host and junction names.

no
When set to no, WebSEAL completely filters the BASE HREF tags. For BASE tags that do not contain a trailing slash WebSEAL strips the last component.

Usage

This stanza entry is optional.

Default value

yes

Example

```
preserve-base-href2 = yes
```

preserve-p3p-policy

Syntax

```
preserve-p3p-policy = {yes|no}
```

Description

Specifies whether to replace or preserve p3p headers from junctioned servers.

Options

yes
The value yes means that headers are preserved.

no
A value of no means that headers are replaced.

Usage

This stanza entry is required.

Default value

no

Example

```
preserve-p3p-policy = no
```

process-root-requests

Syntax

```
process-root-requests = {never|always|filter}
```

Description

Specifies how WebSEAL responds to requests for resources located at the root ("/") junction.

Options

never
Root junction requests are never processed at the root junction.

always
Always attempt to process requests for the root junction at the root junction first before attempting to use a junction mapping mechanism.

filter
Examine all root junction requests to determine whether they start with the patterns specified in the **[process-root-filter]** stanza.

Usage

This stanza entry is required.

Default value

`always`

Example

```
process-root-requests = always
```

proxy-expect-header

Use this entry to control whether the ‘expect: 100-continue’ header is handled natively, or whether it is forwarded to the junctioned server.

Syntax

```
proxy-expect-header = {yes|no}
```

Description

If a server receives an Expect HTTP request header, with a value of '100-continue', it is expected to respond with either a 417 (Expectation Failed) or 100 (Continue) status. WebSEAL, as per the specification, will by default forward the request onto the junctioned server and allow the junctioned server to respond with a 417 or 100. In the event that the junctioned server does not support the 'Expect' header this configuration entry, when set to 'no', will force WebSEAL to respond with a 100 status on behalf of the junctioned server.

Options

yes

The ‘expect: 100-continue’ header is proxied to the junctioned server for processing.

no
The 'expect: 100-continue' header is handled locally.

Usage

This stanza entry is not required.

Default Value

yes

Example

```
proxy-expect-header = yes
```

redirect-using-relative

Syntax

```
redirect-using-relative = {true|false}
```

Description

Specifies that WebSEAL use a server-relative format for the URL in the **Location** header of an HTTP 302 redirect response.

This configuration change affects all redirect responses generated by WebSEAL. These redirect situations include:

- Redirect after authentication
- Redirect after logout
- Redirect after changing password
- Redirects during the e-community single signon authentication process
- Redirects during the cross-domain single signon authentication process
- Switch user processing
- Certificate authentication (**prompt-as-needed** only)
- Session displacement

Options

true
Use a server-relative format for the URL in the **Location** header of an HTTP 302 redirect response.

false
Use an absolute format for the URL in the **Location** header of an HTTP 302 redirect response.

Usage

This stanza entry is not required and is a hidden entry.

Default value

`false`

Example

```
redirect-using-relative = true
```

reject-invalid-host-header

Syntax

```
reject-invalid-host-header = {yes|no}
```

Description

Determines whether requests to WebSEAL that have an invalid host header (see RFC2616) are rejected with a status of 400, "Bad Request."

Options

yes
All requests to WebSEAL with an invalid host header will be rejected with a status of 400, "Bad Request."

no
Requests with an invalid host header are not rejected.

Usage

This stanza entry is required.

Default value

no

Example

```
reject-invalid-host-header = no
```

reject-request-transfer-encodings

Syntax

```
reject-request-transfer-encodings = {yes|no}
```

Description

Specifies the WebSEAL response to requests containing the Transfer-Encoding header.

Options

yes
WebSEAL rejects (with error status of 501, Not Implemented) any request with a Transfer-Encoding header value of anything other than "identity" or "chunked".

no
WebSEAL may reject the request, or may forward it on the junctioned server in a corrupted state. This setting is available for compatibility with versions of WebSEAL prior to version 6.0.

Usage

This stanza entry is required.

Default value

yes

Example

```
reject-request-transfer-encodings = yes
```

request-body-max-read

Syntax

```
request-body-max-read = number_of_bytes
```

Description

Maximum number of bytes to read in as content from the body of POST requests. The request-body-max-read stanza entry affects the request body only. It does not impose limits on other components of a request, such as request line and headers. Used for dynurl, authentication, and request caching.

Options

number_of_bytes
Maximum number of bytes to read in as content from the body of POST requests. Used for dynurl, authentication, and request caching. Minimum number of bytes: 512.

Usage

This stanza entry is required.

Default value

4096

Example

```
request-body-max-read = 4096
```

request-max-cache

Syntax

```
request-max-cache = number_of_bytes
```

Description

Maximum amount of data to cache. This is used to cache request data when a user is prompted to authenticate before a request can be fulfilled.

Options

number_of_bytes
This value should be a positive integer. If set to zero (0), the user login succeeds but the request fails because WebSEAL cannot cache the request data. There is no maximum value.

Usage

This stanza entry is required.

Default value

8192

Example

```
request-max-cache = 8192
```

redirect-http-to-https

Use the `redirect-http-to-https` stanza entry to control whether WebSEAL redirects HTTP requests to HTTPS.

Syntax

```
redirect-http-to-https = {yes | no}
```

Description

When this configuration entry is set to `yes`, WebSEAL automatically redirects HTTP requests to the corresponding HTTPS resource. If no corresponding HTTPS interface is available the HTTP request will proceed as per normal.

Options

yes
When `redirect-http-to-https` is set to `yes`, WebSEAL redirects HTTP requests to HTTPS.

no
When `redirect-http-to-https` is set to `no`, WebSEAL does not redirect HTTP requests to HTTPS.

Usage

This stanza entry is optional.

Default value

no

Example

```
redirect-http-to-https = yes
```

send-header-ba-first

Syntax

```
send-header-ba-first = {yes|no}
```

Description

By default, WebSEAL selects the authentication challenge to return to the client by sequentially searching the available authentication mechanisms until it finds one that is enabled. You can use the **send-header-ba-first** entry to ensure that WebSEAL selects the BA header before any of the other configured authentication mechanisms.

Options

yes
WebSEAL sends the header first.

no
WebSEAL searches sequentially through the available authentication mechanisms and sends the first one that is enabled.

Usage

This stanza entry is optional.

Default value

no

Example

```
send-header-ba-first = yes
```

See also

“send-header-spnego-first” on page 442

send-header-spnego-first

Syntax

```
send-header-spnego-first = {yes|no}
```

Description

By default, WebSEAL selects the authentication challenge to return to the client by sequentially searching the available authentication mechanisms until it finds one that is enabled. You can use the **send-header-spnego-first** entry to ensure that WebSEAL selects SPNEGO header first before any of the other configured authentication mechanisms.

SPNEGO authentication can use either forms login or a header.

Note: If **send-header-ba-first** is set to yes and **send-header-spnego-first** is set to no, WebSEAL sends a BA header first, but uses the default search for an SPNEGO forms login.

Options

yes
WebSEAL sends the header first.

no
WebSEAL searches sequentially through the available authentication mechanisms and sends the first one that is enabled.

Usage

This stanza entry is optional.

Default value

no

Example

```
send-header-spnego-first = yes
```

See also

“send-header-ba-first” on page 442

server-name

Syntax

```
server-name = host_name-instance_name
```

Description

The WebSEAL instance name.

Options

host_name-instance_name
The WebSEAL instance name, based on the host name of the machine and the instance name of the WebSEAL server. This value is set by the administrator during WebSEAL configuration. WebSEAL instance names must be alphanumeric. The maximum number of characters allowed is 20.

Usage

This stanza entry is required.

Default value

None.

Example

Example initial WebSEAL server with the default instance name accepted, on a host named `diamond`:

```
server-name = diamond-default
```

Example instance WebSEAL instance, specified as **web2**, on a host named diamond:

```
server-name = diamond-web2
```

slash-before-query-on-redirect

Syntax

```
slash-before-query-on-redirect = {yes|no}
```

Description

When a client URL specifies a directory location that does not end in a trailing slash (/), the client is redirected to the same URL with a trailing slash added.This is necessary for ACL checks to work properly.

This stanza entry controls where the slash is added if the original URL contains a query string.

Options

yes
Setting this value to yes causes the trailing slash to be added *before* the query string.

For example: /root/directoryname?query
becomes /root/directoryname/?query

no
Setting this value to no causes the trailing slash to be added *after* the query string.

For example: /root/directoryname?query
becomes /root/directoryname?query/

NOTE: A setting of no could cause browser errors. This option exists for backwards compatibility only.

Usage

This stanza entry is optional.

Default value

no

Example

```
slash-before-query-on-redirect = yes
```

strip-www-authenticate-headers

Syntax

```
strip-www-authenticate-headers = {yes|no}
```

Description

Controls whether WebSEAL removes the following headers from the responses that it receives from junctioned servers:

- **Negotiate www-authenticate** header.

- **NTLM www-authenticate** header.

Options

yes
When set to yes, WebSEAL removes these **www-authenticate** headers from junctioned server responses.

no
When set to no, WebSEAL does not remove these **www-authenticate** headers from junctioned server responses.

Usage

This stanza entry is optional.

Default value

yes

Example

```
strip-www-authenticate-headers = yes
```

suppress-backend-server-identity

Syntax

```
suppress-backend-server-identity = {yes|no}
```

Description

Suppresses the identity of the back-end application server from HTTP responses. These responses normally include the line:

```
Server: IBM_HTTP_SERVER/version_number Apache/version_number (Win32)
```

Options

yes
Setting this value to yes deletes the above header line from the server response.

no
Setting this value to no leaves the above header line in the server response.

Usage

This stanza entry is required.

Default value

no

Example

```
suppress-backend-server-identity = no
```

suppress-dynurl-parsing-of-posts

Syntax

```
suppress-dynurl-parsing-of-posts = {yes|no}
```

Description

Determines whether POST bodies are used in dynurl processing.

Note: Before enabling this option, make certain that no dynurl checked server applications accept arguments from POST bodies so that dynurl checks cannot be bypassed using a POST instead of a Query string.

Options

yes
POST bodies will not be used in dynurl processing, only Query strings will be used.

no
POST bodies can be used in dynurl processing.

Usage

This stanza entry is required.

Default value

no

Example

```
suppress-dynurl-parsing-of-posts = no
```

suppress-server-identity

Syntax

```
suppress-server-identity = {yes|no}
```

Description

Suppresses the identity of the WebSEAL server from HTTP responses. These responses normally include the line:

```
Server: WebSEAL/version_number
```

Options

yes
Setting this value to yes deletes the above header line from the server response.

no
Setting this value to no leaves the above header line in the server response.

Usage

This stanza entry is required.

Default value

yes

Example

```
suppress-server-identity = yes
```

tag-value-missing-attr-tag

Syntax

```
tag-value-missing-attr-tag = tag_for_missing_attribute
```

Description

WebSEAL allows credential attributes to be inserted into the HTTP stream as HTTP headers. In the event that a requested attribute is not found in the credential, the HTTP header is still created with a static string. The `tag-value-missing-attr-tag` configuration entry defines the contents of the header.

Note: The `tag-value-missing-attr-tag` is only visible to authenticated users.

Options

tag_for_missing_attribute
Tag inserted in the HTTP header in place of a missing attribute.

Usage

This stanza entry is required.

Default value

NOT_FOUND

Example

```
tag-value-missing-attr-tag = NOT_FOUND
```

update-content-cache-stale-entries-only

The `update-content-cache-stale-entries-only` configuration entry controls whether cached entries are only replaced when they are stale, or whether they are always replaced with the latest response received from the server.

Syntax

```
update-content-cache-stale-entries-only = {true|false}
```

Description

Use this configuration entry to control whether cached entries are only replaced when they are stale, or whether they are always replaced with the latest response received from the server. The server will receive an updated response from the server for a cached resource when the '**cache-control**' request header has been set to 'no-cache'.

Options

true
Set the configuration entry to true if only stale cache entries are to be replaced.

false
Set the configuration entry to false if cached entries are always replaced with the latest response received from the server.

Usage

This stanza entry is optional.

Default value

false

Example

```
update-content-cache-stale-entries-only = false
```

use-existing-username-macro-in-custom-redirects

Syntax

```
use-existing-username-macro-in-custom-redirects  = {yes|no}
```

Description

When using Local Response Redirection, you can use this configuration option to control how WebSEAL processes the USERNAME macro. By default, WebSEAL sets the USERNAME macro value to the string "unauthenticated" after an inactivity timeout. This processing does not match the behavior when WebSEAL serves static pages.

Use this option to override the default behavior and configure WebSEAL to set the USERNAME macro value to the authenticated username. That is, with this option set to yes, WebSEAL processes the USERNAME macro the same when using Local Response Redirection as it does when serving static pages.

Options

yes
When using Local Response Redirection, the USERNAME macro value is set to the authenticated username after an inactivity timeout.

no
When using Local Response Redirection, the USERNAME macro value is set to the string "unauthenticated" after an inactivity timeout.

Usage

This stanza entry is optional.

Default value

no

Example

```
use-existing-username-macro-in-custom-redirects  = yes
```

use-http-only-cookies

Syntax

```
use-http-only-cookies = {yes|no}
```

Description

Indicates whether WebSEAL will add the HTTP-only attribute to the Session, LTPA and Failover Set-Cookie headers sent by WebSeal.

Options

yes
Enables WebSEAL to add the HTTP-only attribute to Session, LTPA and Failover Set-Cookie headers.

no
Prevents WebSEAL from adding the HTTP-only attribute to Session, LTPA and Failover Set-Cookie headers.

Usage

This stanza entry is required.

Default value

yes

Example

```
use-http-only-cookies = yes
```

utf8-form-support-enabled

Syntax

```
utf8-form-support-enabled = {yes|no|auto}
```

Description

UTF-8 encoding support.

Options

yes
WebSEAL only recognizes UTF-8 encoding in forms and the data is used without modification.

no
WebSEAL does not recognize UTF-8 encoding in forms. Used for local code page only.

auto
When set to `auto`, WebSEAL attempts to distinguish between UTF-8 and other forms of language character encoding. When encoding is not recognized as UTF-8, WebSEAL processes the coding as non-UTF-8.

Usage

This stanza entry is required.

Default value

`yes`

Example

```
utf8-form-support-enabled = yes
```

utf8-qstring-support-enabled

Syntax

```
utf8-qstring-support-enabled = {yes|no|auto}
```

Description

UTF-8 encoding support.

Options

yes
WebSEAL only recognizes UTF-8 encoding in strings and the data is used without modification.

no
WebSEAL does not recognize UTF-8 encoding in strings. Used for local code page only.

auto
When set to `auto`, WebSEAL attempts to distinguish between UTF-8 and other forms of language character encoding. When encoding is not recognized as UTF-8, WebSEAL processes the coding as non-UTF-8.

Usage

This stanza entry is required.

Default value

no

Example

```
utf8-qstring-support-enabled = no
```

utf8-url-support-enabled

Syntax

```
utf8-url-support-enabled = {yes|no|auto}
```

Description

Enable or disable support for UTF-8 encoded characters in URLs.

Options

yes
WebSEAL only recognizes UTF-8 encoding in URLs and the data is used without modification.

no
WebSEAL does not recognize UTF-8 encoding in URLs. Used for local code page only.

auto
When set to `auto`, WebSEAL attempts to distinguish between UTF-8 and other forms of language character encoding. When encoding is not recognized as UTF-8, WebSEAL processes the coding as non-UTF-8.

Usage

This stanza entry is required.

Default value

`yes`

Example

```
utf8-url-support-enabled = yes
```

validate-query-as-ga

Syntax

```
validate-query-as-ga = {yes|no}
```

Description

Determines whether WebSEAL returns a "Bad Request" error when there is an invalid character present in the query portion of the URL.

Options

yes
WebSEAL does not return a "Bad request" error when there is an invalid character present in the query portion of the URL.

no
WebSEAL returns a "Bad Request" error when there is an invalid character present in the query portion of the URL.

Usage

This stanza entry is optional.

Default value

no

Example

```
validate-query-as-ga = yes
```

web-host-name

Syntax

```
web-host-name = manually-set-webseal-hostname
```

Description

The manual setting for the WebSEAL server's host name.If left unset, WebSEAL attempts to automatically determine the server's host name. On systems with many hostnames, interfaces, or WebSEAL instances, the automatic determination may not always be correct. The manual setting for **web-host-name** resolves any conflicts.

Options

manually-set-webseal-hostname
The manual setting for the WebSEAL server's host name, based on the fully qualified machine name.

Usage

This stanza entry is optional.

Default value

`www.webseal.com`

Example

```
web-host-name = abc.example.com
```

web-http-port

Syntax

```
web-http-port = port for web-http-protocol
```

Description

Defines the port that the client Web browser uses to connect to WebSEAL for requests that WebSEAL receives on a TCP interface.

Options

port for web-http-protocol

Usage

This stanza entry is optional.

Default value

same as HTTP port

Example

```
web-http-port = 443
```

web-http-protocol

Syntax

```
web-http-protocol = {http | https}
```

Description

Defines the protocol that the client Web browser uses to connect to WebSEAL for requests that WebSEAL receives on a TCP interface.

Options

http
WebSEAL functions will behave as if the client is connected to WebSEAL in an HTTP environment (*not* HTTPS).

https
Most WebSEAL functions will behave as if the client is connected to WebSEAL in an HTTPS environment. There are exceptions and limitations to this rule. You cannot obtain SSL IDs or SSL client certificates using this parameter; therefore, `[session] ssl-id-sessions` cannot be used as a session key and `[certificate] accept-client-certs` cannot be used for authentication.

Usage

This stanza entry is optional.

Default value

`http`

Example

```
web-http-protocol = http
```

web-https-port

Syntax

```
web-https-port = port for web-https-protocol
```

Description

Defines the port that the client Web browser uses to connect to WebSEAL for requests that WebSEAL receives on an SSL interface.

Options

port for web-https-protocol

Usage

This stanza entry is optional.

Default value

same as https port

Example

```
web-https-port = 443
```

web-https-protocol

Syntax

```
web-https-protocol = {http | https}
```

Description

Defines the protocol that the client Web browser uses to connect to WebSEAL for requests that WebSEAL receives on an SSL interface.

Options

http
WebSEAL functions will behave as if the client is connected to WebSEAL in an HTTP environment (*not* HTTPS).

https
Most WebSEAL functions will behave as if the client is connected to WebSEAL in an HTTPS environment. There are exceptions and limitations to this rule. You cannot obtain SSL IDs or SSL client certificates using this parameter; therefore, `[session] ssl-id-sessions` cannot be used as a session key and `[certificate] accept-client-certs` cannot be used for authentication.

Usage

This stanza entry is optional.

Default value

https

Example

```
web-https-protocol = https
```

worker-threads

Syntax

```
worker-threads = number_of_threads
```

Description

Number of WebSEAL worker threads.

Options

number_of_threads
Number of WebSEAL worker threads. The minimum value is 1. The maximum number of threads is based on the number of file descriptors set for WebSEAL at compile time. Note that this number varies per operating system. If the value is set to a number larger than the WebSEAL-determined limit, WebSEAL reduces the value to the acceptable limit and issues a warning message.

Usage

This stanza entry is required.

Default value

300

Example

```
worker-threads = 300
```

[server:<jct-id>] stanza

auth-challenge-type

Use the **auth-challenge-type** stanza entry to specify a comma-separated list of authentication types that WebSEAL can use to challenge a client for authentication information.

Syntax

```
auth-challenge-type = list
```

Description

Each authentication type can be customized for particular user agent strings. For more information about authentication challenges based on the user agent, see the *IBM Security Verify Access: Web Reverse Proxy Configuration Guide*.

You can customize this configuration item for a particular junction by adding the adjusted configuration item to a **[server:*{jct_id}*]** stanza.

where *{jct-id}* refers to the junction point for a standard junction (including the leading / character) or the virtual host label for a virtual host junction.

Options

list

A comma-separated list of authentication types that is used when challenging a client for authentication information. The supported authentication types include:

- ba
- cert
- eai
- forms
- oidc
- spnego

The corresponding authentication configuration entry (for example, ba-auth) must be enabled for each specified authentication challenge type.

Each authentication type can also be qualified with a set of rules to specify the user agents that receive a given challenge type. These rules are separated by semicolons and placed inside square brackets preceding the authentication type. Each rule consists of a plus (+) or minus (-) symbol to indicate inclusion or exclusion, and the pattern to match on. The pattern can include:

- Alphanumeric characters
- Spaces
- Periods (.)
- Wildcard characters, such as, question mark (?) and asterisk (*)

Usage

This stanza entry is optional.

Default value

By default, the list of authentication challenge types matches the list of configured authentication mechanisms.

```
auth-challenge-type = ba
auth-challenge-type = forms
```

Example

```
auth-challenge-type = ba, forms
auth-challenge-type = [-msie;+ms]ba, [+mozilla*;+*explorer*]forms
```

[session] stanza

client-identifier

Use this entry to designate the client identifier for the session.

Syntax

```
client-identifier = client_identifier_value
```

Description

This identifier is added to the credential as the **client_identifier** attribute and is validated on subsequent requests to ensure that the client does not change.

Options

client_identifier_value

The client identifier value can be in either of the following formats:

CLIENT_IP

The client IP address from the network connection is used as the identifier.

HTTPHDR{*<name>*}

The contents of the HTTP header, which is identified by *<name>*, is used as the client identifier. If the HTTP header is missing on the initial request, no identifier is added for the session.

Usage

This stanza entry is optional.

Note: If failover cookies are used, add the **client_identifier** credential attribute to the failover cookie by modifying the **[failover-add-attributes]** and **[failover-restore-attributes]** stanzas. This step ensures that the client identifier can persist across a failover event.

Default value

None

Example

```
client-identifier = CLIENT_IP
client-identifier = HTTPHDR{X-Forwarded-For}
```

create-unauth-sessions

Use this entry to define whether to establish sessions for access to unprotected resources.

Syntax

```
create-unauth-sessions = {yes|no}
```

Description

This configuration entry is useful when a consistent session identifier is required for clients as they make the transition from unauthenticated to authenticated.

Options

yes
Establish sessions for access to unprotected resources.

no
Do not establish sessions for access to unprotected resources.

Usage

This stanza entry is optional.

Default value

```
no
```

Example

```
create-unauth-sessions = no
```

dsess-auto-update

Use the **dsess-auto-update** stanza entry to define whether the Distributed Session Cache (DSC) configuration will be automatically updated if the DSC configuration within a Docker environment changes.

Syntax

```
dsess-auto-update = {yes|no}
```

Description

Enable or disable auto update of the DSC configuration in a Docker environment.

Note: This configuration entry only applies to a Docker environment and does not impact a traditional appliance environment.

Options

yes
Enable auto update of the DSC configuration in a Docker environment.

no
Disable auto update of the DSC configuration in a Docker environment.

Usage

This stanza entry is optional.

Default value

yes

Example

```
dsess-auto-update = yes
```

dsess-enabled

Use the **dsess-enabled** stanza entry to enable or disable the use of the distributed session cache.

Syntax

```
dsess-enabled = {yes|no}
```

Description

Enable or disable use of the distributed session cache.

Options

yes
Enable use of the distributed session cache. If this is set to yes the **[dsess]** stanza must have information about how to communicate with the distributed session cache.

no
Disable use of the distributed session cache.

Usage

This stanza entry is optional.

Default value

no

Example

```
dsess-enabled = no
```

dsess-last-access-update-interval

Use the **dsess-last-access-update-interval** stanza entry to specify the frequency, in seconds, at which WebSEAL updates the session last access time at the distributed session cache.

Syntax

```
dsess-last-access-update-interval = seconds
```

Description

Specifies the frequency at which WebSEAL updates the session last access time at the distributed session cache.

Options

seconds
Smaller values offer more accurate inactivity timeout tracking, at the expense of sending updates to the distributed session cache more frequently. Values of less than 1 second are not permitted.

Usage

This stanza entry is required.

Default value

60

Example

```
dsess-last-access-update-interval = 60
```

dsess-server-type

The `dsess-server-type` configuration entry specifies the type of server which is used to store distributed sessions.

Syntax

```
dsess-server-type = {dsc|redis}
```

Description

The type of server which is used to store distributed sessions.

Options

dsc
The sessions are managed by the Distributed Session Cache (DSC) server. The `[dsess]` stanza must be configured with information on how to communicate with the DSC.

redis
The sessions are managed by a Redis server. The `[redis]` stanza must be configured with information on how to communicate with the Redis server.

Usage

This stanza entry is optional

Default value

None

Example

```
dsess-server-type = dsc
```

dsess-support-local-sessions

Use the `dsess-support-local-sessions` stanza entry to control whether non-cookie based sessions can be stored locally when the distributed session cache is enabled.

Syntax

```
dsess-support-local-sessions = {yes|no}
```

Description

The distributed session cache is dependent on the use of session cookies in order to index into the session cache. This configuration entry controls whether an additional local-only session cache is made available for non-cookie based sessions. An example of a non-cookie based session is an OAuth authenticated session where the authorization header is used as an index into the session cache. By

enabling this configuration entry the server is able to store these types of sessions in a local-only session cache.

Options

Yes
If the distributed session cache is enabled, allow the use of a local session cache for non-cookie based sessions.

No
If the distributed session cache is enabled, do not allow the use of a local session cache for non-cookie based sessions.

Usage

This stanza entry is optional.

Default Value

No

Example

```
dsess-support-local-sessions = no
```

enforce-max-sessions-policy

Use the **enforce-max-sessions-policy** stanza entry to control whether a specific WebSEAL instance enforces the `max-concurrent-web-sessions` policy. The `max-concurrent-web-sessions` policy controls the number of sessions each user can have at one time in a distributed session cache environment.

Syntax

```
enforce-max-sessions-policy = {yes|no}
```

Description

Control whether or not a specific WebSEAL instance enforces the `max-concurrent-web-sessions` policy.

Options

yes
Enforce the `max-concurrent-web-sessions` policy.

no
Do not enforce the `max-concurrent-web-sessions` policy.

Usage

This stanza entry is ignored unless WebSEAL is using the distributed session cache for session storage.

Default value

`yes`

Example

```
enforce-max-sessions-policy = yes
```

inactive-timeout

Syntax

```
inactive-timeout = number_of_seconds
```

Description

Integer value for lifetime, in seconds, of inactive entries in the credential cache.

The value can be configured for a specific session cache (authenticated or unauthenticated) by adding an additional entry, prefixedby `auth` or `unauth`.

Options

number_of_seconds
The minimum number for this value is 0. WebSEAL does not impose a maximum value.

A stanza entry value of "0" disables this inactivity timeout feature (inactivity timeout value is unlimited). The control of cache entries is then governed by the **timeout** and **max-entries** stanza entries.

When a cache is full, the entries are cleared based on a least-recently-used algorithm.

Usage

This stanza entry is required.

Default value

`600`

Example

```
inactive-timeout = 600
unauth-inactive-timeout = 300
```

logout-remove-cookie

Syntax

```
logout-remove-cookie = {yes|no}
```

Description

Specifies whether or not to remove the session cookie from a user's browser when the user logs out from the WebSEAL domain. Setting this stanza entry to `yes` is necessary for the correct operation and use of the %OLDSESSION% macro.

Options

yes
Remove the session cookie from a user's browser when the user logs out from the WebSEAL domain.

no
Do not remove the session cookie from a user's browser when the user logs out from the WebSEAL domain.

Usage

This stanza entry is required.

Default value

no

Example

```
logout-remove-cookie = no
```

max-entries

Syntax

```
max-entries = number_of_entries
```

Description

Maximum number of concurrent entries in the credentials cache. When the cache size reaches this value, entries are removed from the cache according to a least recently used algorithm to allow new incoming logins.

The value can be configured for a specific session cache (authenticated or unauthenticated) by adding an additional entry, prefixedby `auth` or `unauth`.

Options

number_of_entries

The following conditions affect the specified value:

- If the specified value is less than or equal to 0, the cache size becomes unlimited.
- If the specified value is between 0 and 8192, the actual number of entries allowed is rounded up to the next multiple of 32.
- Any specified value greater than 8192 is accepted as given.

WebSEAL does not impose a maximum value.

Usage

This stanza entry is required.

Default value

4096

Example

```
max-entries = 4096
unauth-max-entries = 1024
```

prompt-for-displacement

Syntax

```
prompt-for-displacement = {yes|no}
```

Description

Determines whether or not a user is prompted for appropriate action when the `max-concurrent-web-sessions` displace policy has been exceeded.

Options

yes
If the max-concurrent-web-sessions policy for the user is set to 'displace', then WebSEAL will prompt the user, using the too_many_sessions.html response page, before automatically displacing the users existing session.

no
If set to "no", and the max-concurrent-web-sessions policy for the user is set to 'displace', then WebSEAL will automatically log out the existing user session. When using the Redis server, and the max-concurrent-web-sessions policy for the user is not set to 'displace', then WebSEAL will automatically log out a random existing session of the user when the concurrent session limit has been reached. A new session will be created for the user and the user is logged in to this new session transparently. The displaced (older) session is no longer valid.

Usage

This stanza entry is required.

Default value

`yes`

Example

```
prompt-for-displacement = yes
```

preserve-inactivity-timeout

Use the **preserve-inactivity-timeout** stanza entry to designate the resources that should not impact the inactivity timeout for the session.

Syntax

```
 preserve-inactivity-timeout = uri
```

Description

In some circumstances, you might not want the requests for a particular resource to affect the inactivity timeout for a session. For example, you might want to preserve the inactivity timeout when a server is polled by an Ajax script running in the background of a client browser. This configuration entry can be used to designate the resources which, when accessed, should not impact the inactivity timeout for the session. A comparison will be performed against either the full HTTP request line or the decoded URI (controlled by the **preserve-inactivity-timeout-match-uri** configuration entry). If a match is

found, the inactivity timeout for the session will not be affected by the request. Multiple patterns can be specified by including multiple configuration entries of the same name.

You also have the option of matching a request by using a host header. This option is useful when you need to selectively enable this functionality for a particular virtual host junction. To selectively match an entry based on a particular host header, prepend the configuration entry with the string `[<host>]`.

Options

uri
The URI to pattern match against.

Usage

This stanza entry is optional.

Default value

Not applicable.

Example

```
preserve-inactivity-timeout = /jct/robot/*
preserve-inactivity-timeout = [www.ibm.com]/robot/*
```

preserve-inactivity-timeout-match-uri

Use this configuration entry to control whether the patterns specified by the **preserve-inactivity-timeout** configuration entry are matched against the decoded URI from the request, or against the full request line.

Syntax

```
preserve-inactivity-timeout-match-uri = {true|false}
```

Description

This configuration entry controls whether the patterns specified by the **preserve-inactivity-timeout** configuration entry are matched against the decoded URI from the request, or against the full request line.

Options

true
The patterns specified by the **preserve-inactivity-timeout** configuration entry are matched against the decoded URI from the request.

false
The patterns specified by the **preserve-inactivity-timeout** configuration entry are matched against the full request line.

Usage

This stanza entry is optional.

Default value

`true`

Example

```
preserve-inactivity-timeout-match-uri = true
```

require-auth-session-http-hdrs

Use the `require-auth-session-http-hdrs` configuration entry to control whether a HTTP header must be used in an authentication operation before it can be used as a session key.

Syntax

```
require-auth-session-http-hdrs = {yes|no}
```

Description

Controls whether a HTTP header must be used in an authentication operation before it can be used as a session key. This helps to ensure that the HTTP header which is used as a session key is secure and reduces the risk of the header being spoofed.

Options

yes
The HTTP header must be used in an authentication operation before it will be accepted as a session key.

no
Any HTTP header can be used as a session key.

Note: If you set this configuration item to no it means that an unauthenticated value could be used as the index into the session. In this situation it is critical to ensure that the client is 'trusted' and is able to produce a secure session key, otherwise you open the environment up to session fixation attacks.

Usage

This stanza entry is optional.

Default value

`yes`

Example

```
require-auth-session-http-hdrs = yes
```

require-mpa

Syntax

```
require-mpa = {yes|no}
```

Description

Controls whether WebSEAL accepts HTTP headers from requests that are proxied through an authenticated multiplexing proxy agent (MPA).

Options

yes
WebSEAL only accepts HTTP headers from requests that are proxied through an authenticated multiplexing proxy agent (MPA).

no
WebSEAL accepts HTTP headers under any condition.

Usage

This stanza entry is required.

Default value

yes

Example

```
require-mpa = yes
```

resend-webseal-cookies

Syntax

```
resend-webseal-cookies = {yes|no}
```

Description

When you configure WebSEAL to use session cookies, specifies whether or not WebSEAL sends the session cookie to the browser with every response.

Options

yes
Specifies that WebSEAL sends the session cookie to the browser with every response. This action helps to ensure that the session cookie remains in the browser memory.

no
Specifies that WebSEAL does not send the session cookie to the browser with every response.

Usage

This stanza entry is required.

Default value

no

Example

```
resend-webseal-cookies = no
```

send-constant-sess

Syntax

```
send-constant-sess = {yes|no}
```

Description

Determines whether a session cookie containing a separate, constant identifier is issued during step-up operations to enable tracking for each authenticated session. The identifier remains constant across a single session, regardless of whether the session key changes. The name of the cookie is that of the actual session code appended with the suffix *-2*, for example, `PD_S_SESSION_ID_2`. This feature is intended to augment the `-k` junction option.

Options

yes
A session cookie containing a separate, constant identifier is issued during step-up operations to allow tracking for each authenticated session.

no
No session cookie is issued during step-up operations.

Usage

This stanza entry is required.

Default value

no

Example

```
send-constant-sess = no
```

shared-domain-cookie

Use the **`shared-domain-cookie`** stanza entry to share a cookie-based session across all standard and virtual host junctions on a single WebSEAL instance.

Syntax

```
shared-domain-cookie = {yes | no}
```

Description

Enables a cookie-based session to be shared across all standard and virtual host junctions on a single WebSEAL instance. To share a session in this manner, the WebSEAL instance must store a single session key as an independent value in a multi-valued domain cookie. The multi-valued domain cookie must be indexed by the instance name.

The domain cookie itself is shared across all participating WebSEAL instances, but the session values are specific to each instance.

If WebSEAL exists in an environment where the distributed session cache already handles single sign-on across domains, do not enable this configuration item.

Options

yes
Enables single sign-on across virtual host junctions in the same WebSEAL instance.

no
Disables single sign-on across virtual host junctions in WebSEAL.

Usage

This stanza entry is optional.

Default value

no

Example

```
shared-domain-cookie = yes
```

ssl-id-sessions

Syntax

```
ssl-id-sessions = {yes|no}
```

Description

Indicates whether to use the SSL ID to maintain a user's HTTP login session.

Options

yes
Use the SSL ID to maintain a user's HTTP login session.

no
Do not use the SSL ID to maintain a user's HTTP login session. This value must be set to no when the following *key = value* pair is set:

```
[certificate]
accept-client-certs = prompt_as_needed
```

Usage

This stanza entry is required.

Default value

yes

Example

```
ssl-id-sessions = yes
```

ssl-session-cookie-name

Syntax

```
ssl-session-cookie-name = name
```

Description

Specifies the default or custom name of WebSEAL session cookies.

Options

name
Specifies the default or custom name of WebSEAL session cookies.

Usage

This stanza entry is required.

Default value

PD-S-SESSION-ID

Example

```
ssl-session-cookie-names = PD-S-SESSION-ID
```

standard-junction-replica-set

Use the **standard-junction-replica-set** stanza entry to specify the replica set to use for sessions that are created when users access standard WebSEAL junctions.

Syntax

```
standard-junction-replica-set = replica_set_name
```

Description

The replica set to use for sessions created when users access standard WebSEAL junctions. Virtual host junctions either use the replica set specified with the **virtualhost create -z** option or the virtual host name for the junction.

If you are using the distributed session cache for session storage, the replica set specified here must also be specified in the **[replica-sets]** stanza.

Options

value
Replica set name.

Usage

This stanza entry is required.

Default value

default

Example

```
standard-junction-replica-set = default
```

tcp-session-cookie-name

Syntax

```
tcp-session-cookie-name = name
```

Description

Specifies the default or custom name of WebSEAL session cookies.

Options

name
Specifies the default or custom name of WebSEAL session cookies.

Usage

This stanza entry is required.

Default value

PD-H-SESSION-ID

Example

```
tcp-session-cookie-name = PD-H-SESSION-ID
```

temp-session-cookie-name

Syntax

```
temp-session-cookie-name = cookie_name
```

Description

Sets the name of the temporary session cookie that is created for session sharing with Microsoft Office applications. WebSEAL creates a temporary cookie with this name when it responds to a **/pkmstempsession** management page request.

Options

cookie_name
A string value that represents the name of the temporary cookie that WebSEAL uses to store session information.

Note: This configuration entry must be used in conjunction with a non-zero value for the **temp-session-max-lifetime** entry, which is also in the **[session]** stanza. For more information about

sharing sessions with Microsoft Office applications, see the *IBM Security Verify Access: Web Reverse Proxy Configuration Guide*.

Usage

This stanza entry is required.

Default value

None.

Example

```
temp-session-cookie-name = PD-TEMP-SESSION-ID
```

temp-session-max-lifetime

Syntax

```
temp-session-max-lifetime = number_of_seconds
```

Description

Positive integer that expresses the maximum lifetime (in seconds) of entries in the temporary session cache.

Options

number_of_seconds
A positive integer that represents the maximum lifetime in seconds. Specify a value of `0` to disable the temporary session cache.

Note: A non-zero value must be configured to enable session sharing with Microsoft Office applications. For more information about sharing sessions with Microsoft Office applications, see the *IBM Security Verify Access: Web Reverse Proxy Configuration Guide*.

Usage

This stanza entry is optional.

Default value

None.

Example

```
temp-session-max-lifetime = 10
```

temp-session-one-time-use

Use the **`temp-session-one-time-use`** stanza entry to control whether the client can access an entry in the temporary session cache a single time only or multiple times. If you are sharing sessions with

Microsoft Office applications, configure WebSEAL to accept multiple accesses to the temporary session cookie.

Syntax

```
temp-session-one-time-use = {true|false}
```

Description

WebSEAL creates a temporary session cookie for session sharing with Microsoft Office applications when it responds to a **/pkmstempsession** management page request.

The **temp-session-one-time-use** configuration entry determines whether this cookie can be used a single time or whether it can be used multiple times.

Note: This configuration entry must be used with the **temp-session-cookie-name** and **temp-session-max-lifetime** entries, which are also in the **[session]** stanza. For more information about sharing sessions with Microsoft Office applications, see the *IBM Security Verify Access: Web Reverse Proxy Configuration Guide*.

Options

true
The cookie that stores the session information is a single use cookie. The session entry is invalidated and removed from the cache after a single use.

false
The client can use the cookie multiple times to request the session information. The session times out after the configured **temp-session-max-lifetime** expires. Use this setting if you want to share sessions with Microsoft Office applications, such as Microsoft SharePoint.

Usage

This stanza entry is required.

Default value

false

Example

```
temp-session-one-time-use = false
```

temp-session-overrides-unauth-session

This stanza entry controls the precedence if both a temporary session cookie and a standard session cookie are provided in a request.

Syntax

```
temp-session-overrides-unauth-session = {yes|no}
```

Description

The **temp-session-overrides-unauth-session** configuration entry controls whether a temporary session cookie takes precedence over a standard session cookie for an unauthenticated session.

Options

yes
A temporary session takes precedence over an existing unauthenticated session, but not an authenticated session.

no
A temporary session cookie is ignored if a standard session cookie exists.

Usage

This stanza entry is optional.

Default value

no

Example

```
temp-session-overrides-unauth-session = no
```

timeout

Syntax

```
timeout = number_of_seconds
```

Description

Integer value for maximum lifetime, in seconds, for an entry in the credential cache.

The value can be configured for a specific session cache (authenticated or unauthenticated) by adding an additional entry, prefixedby `auth` or `unauth`.

Options

number_of_seconds
The minimum number for this value is 0. WebSEAL does not impose a maximum value.

A stanza entry value of "0" disables this timeout feature (lifetime value is unlimited). The control of cache entries is then governed by the **inactive-timeout** and **max-entries** stanza entries.

When the cache is full, the entries are cleared based on a least-recently-used algorithm.

Usage

This stanza entry is required.

Default value

`3600`

Example

```
timeout = 3600
unauth-timeout = 600
```

update-session-cookie-in-login-request

Syntax

```
update-session-cookie-in-login-request = {yes|no}
```

Description

Controls whether the existing session cookie, found in the HTTP request, is updated if the session ID is modified during the processing of the request.

Options

yes

The existing session cookie is updated if the session ID is modified during the processing of the request.

no

The existing session cookie is not updated if the session ID is modified during the processing of the request.

Usage

This stanza entry is optional.

Default value

no

Example

```
update-session-cookie-in-login-request = no
```

user-identity-attribute-name

Use this entry to designate the credential attribute which uniquely identifies the user.

Syntax

```
user-identity-attribute-name = <attribute-name>
```

Description

The name of the credential attribute which holds the unique user identity for the session. This is used, for example, when enforcing the maximum concurrent session policy. If the configured attribute does not exist in the credential the default user identity of 'unknown' will be used. If the configuration entry is not specified, the contents of the `AZN_CRED_PRINCIPAL_NAME` attribute will be used to uniquely identify the user.

Options

<attribute-name>
The name of the credential attribute which uniquely identifies the user.

Usage

This stanza entry is optional.

Default Value

AZN_CRED_PRINCIPAL_NAME

Example

```
user-identity-attribute-name = AZN_CRED_PRINCIPAL_NAME
```

user-session-ids

Syntax

```
user-session-ids = {yes|no}
```

Description

Enables or disables the creation and handling of user session IDs.

Options

yes

Enables the creation and handling of user session IDs.

no

Disables the creation and handling of user session IDs.

Usage

This stanza entry is required.

Default value

no

Example

```
user-session-ids = yes
```

user-session-ids-include-replica-set

Syntax

```
user-session-ids-include-replica-set = {yes|no}
```

Description

Include the replica set in the user session ID.

Options

yes
If set to "yes", then `user-session-ids = yes` includes the replica set.

no
If set to "no", then WebSEAL does not include the replica set for `user-session-ids = yes` and assumes that any user session specified in the **pdadmin terminate session** command belongs to the default replica set.

Usage

This stanza entry is required.

Default value

`yes`

Example

```
user-session-ids-include-replica-set = yes
```

use-same-session

Syntax

```
use-same-session = {yes|no}
```

Description

Indicates whether to use the same session for SSL and HTTP clients.

Options

yes
When set to `yes`, a user who has authenticated over HTTP will be authenticated when connecting over HTTPS. Likewise, the user who has authenticated over HTTPS will be authenticated when connecting over HTTP. Using `yes` will override `ssl-id-sessions = yes`, because HTTP clients do not read an SSL ID to maintain sessions.

no
Do not use the same session for SSL and HTTP clients.

Usage

This stanza entry is required.

Default value

no

Example

```
use-same-session = no
```

[server:<instance>] stanza

This stanza defines a back-end LDAP server that can be used for federated registries.

All operations to specific branches of the LDAP DIT can be passed through to this back-end server.

You can use multiple instances of this stanza.

[ldap] configuration entries that are not configurable in this stanza use the [ldap] stanza values. For example, "`tls-v12-enable`".

For Active Directory back-end, you must use an SSL connection (ssl-enabled = yes, port = 636).

bind-auth-and-pwdchg

Use the **`bind-auth-and-pwdchg`** stanza entry to control whether to force authentication to use the LDAP bind operation and force password change.

Syntax

```
bind-auth-and-pwdchg = {yes | no}
```

Description

This option, when set to `yes`, forces authentication to use the LDAP bind operation rather than the LDAP compare operation. It also forces password change (not reset) to occur on a connection to the LDAP server that is bound as the user that is being changed. An example of a password change is the use of **`/pkmspasswd`** from WebSEAL.

If this option is set to `yes`, then the LDAP server must allow users to change their own password. But in many cases, allowing user to change their own password is not the default behavior. For example, the IBM Security Directory Server requires an ACL to be set in the Directory Information Tree (DIT) for the affected users. Here is example of an ACL that can be inherited, which allows any user to change their own password:

```
aclEntry: access-id:cn=this:at.userPassword:grant:w
```

Options

yes
Force LDAP bind operation and password change.

no
Do not force LDAP bind operation and password change.

Usage

This stanza entry is optional.

Default value

The default value is no.

Example

```
bind-auth-and-pwdchg = yes
```

bind-dn

Use the **bind-dn** stanza entry to define the LDAP user distinguished name (DN) that is used for signing on to the LDAP server.

Syntax

```
bind-dn = LDAP_DN
```

Description

LDAP user distinguished name (DN) that is used when binding (or signing on) to the LDAP server.

Options

LDAP_DN
LDAP user distinguished name (DN) that is used for signing on to the LDAP server.

Usage

This stanza entry is required.

Default value

None.

Example

```
bind-dn = CN=Administrator,CN=Users,DC=ibm,DC=com
```

bind-pwd

Use the **bind-pwd** stanza entry to define password for the distinguished name declared in the **bind-dn** stanza entry.

Syntax

```
bind-pwd = LDAP_password
```

Description

Password for the LDAP user distinguished name declared in the **bind-dn** stanza entry.

Options

LDAP_password
Password for the LDAP user distinguished name declared in the **bind-dn** stanza entry.

Usage

This stanza entry is required and stored in the `ldap.conf` obfuscated file.

Default value

None.

Example

```
bind-pwd = zs77WVoLSZn1rKrL
```

dn-map

Use the **dn-map** stanza entry to define which areas of the Security Verify Access registry have copies of users from this back-end server.

Syntax

```
dn-map = from_dn | to_dn
```

Description

The **dn-map** entries define which areas of the Security Verify Access registry have copies of users from this back-end server. It is only used to pass through password operations.

Options

from_dn
Defines the Security Verify Access registry location of the users copies. This value must be unique across all back-end servers.

to_dn
Defines the back-end registry location of the real users.

Usage

This stanza entry is optional.

The values must be as specific (longest matching) as possible to contain their matches to include only branches of LDAP that are relevant.

Change any value that contains a "|" character to "||" so that it is not misinterpreted as the separator character. The "||" character is reverted to the "|" character before use.

Multiple **dn-map** values can be provided per back-end server.

The most specific (longest matching) **dn-map** is selected. So overlapping maps can be defined.

Multiple entries are allowed.

Default value

None.

Example

The back-end users are all found under the LDAP location:

```
cn=Users|Groups,o=ibm,c=us
```

and they are replicated to the Security Verify Access registry at:

```
cn=Users|Groups,dc=iswga
```

Then the **dn-map** entry would be:

```
dn-map = cn=Users||Groups,dc=iswga | cn=Users||Groups,o=ibm,c=us
```

Thus the Security Verify Access registry user DN of:

```
cn=Test User,cn=Users|Groups,dc=iswga
```

would map to the back-end server user DN of:

```
cn=Test User,cn=Users|Groups,o=ibm,c=us
```

dynamic-groups-enabled

Use the **dynamic-groups-enabled** stanza entry to control whether dynamic groups are supported.

Syntax

```
dynamic-groups-enabled = {yes|no}
```

Description

Indication of whether dynamic groups are supported. This key value pair applies to supported LDAP registries. Security Verify Access supports dynamic groups with IBM Security Directory Server regardless of this setting.

Options

yes
Security Verify Access attempts to resolve dynamic group membership.

no
Security Verify Access does not attempt to resolve dynamic group membership. Anything other than yes, including a blank value, is interpreted as no.

Usage

This stanza entry is optional.

Default value

The default value is no.

Example

```
dynamic-groups-enabled = no
```

group-membership-search-filter

Use the **group-membership-search-filter** entry to specify the LDAP search filter that is used by Security Verify Access to obtain the group membership for a user.

Syntax

```
group-membership-search-filter = ldap search filter
```

Description

Use this configuration file parameter to specify how to locate Security Verify Access group membership for a user in LDAP.

Options
Specifies the LDAP search filter that is used by Security Verify Access to locate group membership for a user in the LDAP directory server. This filter must be a valid LDAP string search filter as described by the Request for Comments (RFC) 2254 document. Any **%dn%** strings found within the filter are replaced with the distinguished name of the user before the search is performed.

Usage

This stanza entry is optional.

Default value

The default value is obtained from the **group-membership-search-filter** entry in the **ldap** stanza. See “group-membership-search-filter” on page 253 .

Example

This example specifies a search for group membership by using the **member** field of objects, with a class of **groupOfNames**, within the LDAP user record.

```
group-membership-search-filter = (&(objectclass=groupOfNames)(member=%dn%))
```

group-search-filter

Use the **group-search-filter** entry to specify the LDAP search filter that is used by Security Verify Access.

Syntax

```
group-search-filter = ldap search filter
```

Description
Use this configuration file parameter to specify how to locate Security Verify Access groups in LDAP.

Options

Specifies the LDAP search filter that is used by Security Verify Access to locate groups in the LDAP directory server. This filter must be a valid LDAP string search filter as described by the Request for Comments (RFC) 2254 document.

Use the **group-search-filter** option with **static-group-objectclass** so that the Security Verify Access can locate LDAP groups that are created with the LDAP object classes.

Do not update the unsupported option with the same name under the **[ldap-generic-general]** stanza.

Usage

This stanza entry is optional.

Default value
Default value is specific to the LDAP server that is detected.

Example

This example specifies a search for a group named `mygroup1` or `mygroup2` under `static-group-objectclass`.

```
group-search-filter = (|(static-group-objectclass=mygroup1)(static-group-objectclass=mygroup2))
```

group-suffix

Use the **group-suffix** stanza entry to define the federated registry suffixes that are used to search for group membership for users.

Syntax

```
group-suffix = dn
```

Description

The **group-suffix** stanza entry defines the federated registry suffixes within the Security Verify Access registry that are searched for group membership for users. All operations on entries under these suffixes occur on the back-end server.

Mapping of the DNs is not possible. So the back-end server must contain a suffix of the same name.

The most specific (longest matching) suffix is selected so that overlapping suffixes can be defined.

Multiple entries are allowed.

Options

dn
Distinguished name of the federated registry LDAP group membership search suffix.

Usage

This stanza entry is optional.

Default value

If no value is specified, the suffixes that are defined by the **suffixes** configuration entry are searched for group membership for users.

Example

```
group-suffix = ou=groups,o=ibm,c=us
```

host

Use the **host** stanza entry to define the host name of the LDAP server.

Syntax

```
host = host_name
```

Description

Host name of the LDAP server.

Options

host_name
Valid values for *host_name* include any valid IP host name. The *host_name* does not have to be a fully qualified domain name.

Usage

This stanza entry is required.

Default value

None.

Example

```
host = diamond
host = diamond.example.com
```

ignore-if-down

Set whether Security Verify Access continues to operate with the other federated registries if one of the registries become unavailable.

Syntax

```
ignore-if-down = { yes | no }
```

Description

Controls whether Security Verify Access ignores this registry when it fails.

Options

yes
Ignore this registry when it fails. Security Verify Access continues to operate with the other federated registries while this registry is unavailable.

Note: If this option is set to yes for a registry, ACLs based on groups in the failed registry will behave as if the group has no members. Duplicate user ID detection is not permitted and must be disabled.

no
Do not ignore this registry when it fails. The registry component of Security Verify Access stops functioning if this registry is unavailable.

Note: For the primary registry, this value cannot be set as primary registry failures cannot be ignored.

Usage

This stanza entry is optional.

Default value

no

Example

```
ignore-if-down = no
```

max-server-connections

Use the **max-server-connections** stanza entry to define the maximum number of connections that are allowed with the LDAP server.

Syntax

```
max-server-connections = number_connections
```

Description

Indicates the maximum number of connections that are allowed with the LDAP server. The Security Verify Access runtime maintains a pool of connections for each LDAP server. From this pool, an available connection is chosen to send requests to the LDAP server. If all connections are busy, a new connection is established with the LDAP server, up to the maximum server connection pool size.

This parameter is not available if you use the **pdconfig** utility. The parameter must be modified manually with the **pdadmin** utility.

Options

number_connections
The maximum number of connections that are allowed with the LDAP server. The valid range for this parameter is 2-16. Values greater than 16 are set to 16.

Usage

This stanza entry is optional.

Default value

If this parameter is not specified, the default pool size is 16.

Example

```
max-server-connections = 16
```

password-attribute

Use this stanza entry to specify the attribute used to set or change passwords.

Syntax

```
password-attribute = name
```

Description

This stanza entry is particularly useful in federated registry environments. It is primarily used to allow RACF suffixes to choose between using "racfpassword" or "racfpassphase".

Options

name
Name of the attribute used to set or change passwords.

Usage

This stanza entry is optional.

Default value

Table 5. Default value of the password-attribute entry

LDAP server	Default value
RACF suffix	`racfpassword`
Active Directory	`unicodePwd`
All others	`userPassword`

Example

```
password-attribute = racfpassword
```

port

Use the **port** stanza entry to define the port appropriate for the **ssl-enabled** value set in this stanza.

Syntax

```
port = port_number
```

Description

Number of the TCP/IP port that is used for communicating with the LDAP server. This port must be the port appropriate for the **ssl-enabled** value set in this stanza.

Options

port_number
A valid port number is any positive integer that is allowed by TCP/IP and that is not currently being used by another application.

Usage

This stanza entry is required.

Default value

None.

Example

```
port = 389
```

pwd-chg-method

This options is used to override the automatically used low level LDAP operations for changing account passwords.

Syntax

```
pwd-chg-method = {automatic|mod_pwd_ext_op|del_add|del_then_add|replace}
```

Description

Overrides the automatically used low level LDAP operations for changing account passwords.

Options

automatic
The LDAP server type is detected and the appropriate LDAP password change operations are selected. This is the default behavior.

mod_pwd_ext_op
The **ldap_extended_operation** 1.3.6.1.4.1.4203.1.11.1 is used.

del_add
A single **ldap_modify** is used that has both a **DELETE** for the old password and an **ADD** for the new password.

del_then_add
Two **ldap_modify** invocations are used. The first does a **DELETE** for the old password. The second does an **ADD** for the new password.

replace
A single **ldap_modify** is used, which does a **REPLACE** of the password attribute with the new password.

Usage

This stanza entry is optional.

Default value

automatic

Example

```
pwd-chg-method = automatic
```

racf-suffix

Use this stanza entry to set whether to treat all suffixes under the **server:<instance>** stanza as RACF suffixes.

Syntax

```
racf-suffix = { yes | no }
```

Description

When this stanza entry is set to "yes", all the suffixes defined under the **server:<instance>** stanza are treated as RACF suffixes.

Take the following points into consideration when you use RACF suffixes:

- RACF suffix users can only be searched for using two attributes: "racfid" and "krbprincipalname". RACF Security Verify Access basic users can only be searched for using the "racfid" and not the "krbprincipalname" attribute.
- It is possible that not all members of a RACF group of type "UNIVERSAL" will be returned. Only the members returned by the group's "racfgroupuserids" attribute will be listed.
- If importing groups or users as full Security Verify Access entities, the primary Security Verify Access registry must provide attribute definitions of all attributes used in the user or group DN. The attributes "profileType" and "racfid" must always be defined as these are always present in RACF user and group DNs. The embedded LDAP server used by Security Verify Access contain definitions for "profileType", "racfid", and "sysplex". External LDAP Security Verify Access primary registries might also need updating to add the missing attribute definitions.
- The RACF suffixes provided must have "profileType=user", "profileType=group", and "profileType=connect" children entries directly under them.
- The pdadmin "user list-dn" and "group list-dn" operations normally use an LDAP filter that matches the "cn" of each user or group entry. For RACF suffixes, it will instead match the "racfid" of the user or group as RACF users do not have a "cn" attribute.
- The pdadmin "user list-dn" and "group list-dn" operations on RACF suffixes will also support the "?" (match any one character) wildcard character as this wildcard cannot be disabled (even though it is non-standard for LDAP search filters).
- A **`user show`** command will display the "racfid" value for "cn" and "sn" as these values do not exist for RACF users.
- The **`user create`** command will ignore the "sn" value provided and will use the "cn" value provided for creating the "racfid" value.
- The **`group create`** command will use the "cn" value provided for creating the "racfid" value.

Options

yes
Treat all suffixes as RACF suffixes.

no
Do not treat all suffixes as RACF suffixes.

Usage

This stanza entry is optional.

Default value

no

Example

```
racf-suffix = no
```

replica

Use the **`replica`** stanza entry to define the LDAP user registry replicas.

Syntax

```
replica = ldap-server, port, type, pref
```

Description

Definition of the LDAP user registry replicas.

Security Verify Access supports a maximum of one host and nine LDAP replica servers. If more than nine LDAP replica entries are listed, the Security Verify Access servers cannot start.

Options

ldap-server
The network name of the server.

port
The port number for the LDAP server. A valid port number is any positive number that is allowed by TCP/IP and that is not currently being used by another application.

type
One of `readonly` or `readwrite`.

pref
A number 1 - 10 (10 is the highest preference).

Usage

This stanza entry is optional.

Default value

None.

Example

Example of one replica specified and two replicas commented out:

```
replica = rep1,390,readonly,1
#replica = rep2,391,readwrite,2
#replica = rep3,392,readwrite,3
```

static-group-objectclass

Use the **`static-group-objectclass`** stanza entry to define the static group object class names to set when you create a native group entry in LDAP.

Syntax

```
static-group-objectclass = object_class_list
```

Description

Contain a list of comma-separated object class names to set when you create a native group entry in LDAP.

Options

object_class_list
A list of comma-separated object class names.

Usage

This stanza entry is optional.

Default value

Default value is specific to the LDAP server that is detected.

Example

```
static-group-objectclass = top,groupOfNames
```

ssl-enabled

Use the **ssl-enabled** stanza entry to enable or disable SSL communication between WebSEAL and the LDAP server.

Syntax

```
ssl-enabled = {yes|no}
```

Description

Enables or disables SSL communication between WebSEAL and the LDAP server.

Options

yes
Enable SSL communication.

no
Disable SSL communication.

Usage

This stanza entry is optional. Ensure that the **port** value is appropriate for the **ssl-enabled** value.

Default value

SSL communication is disabled by default.

Example

```
ssl-enabled = yes
```

ssl-keyfile-dn

Use the **ssl-keyfile-dn** stanza entry to define the key label of the client personal certificate within the SSL key file.

Syntax

```
ssl-keyfile-dn = key_label
```

Description

String that specifies the key label of the client personal certificate within the SSL key file. This key label is used to identify the client certificate that is presented to the LDAP server.

Options

key_label
String that specifies the key label of the client personal certificate within the SSL key file.

Usage

This stanza entry is optional. The certificate that is referenced by this stanza entry is from the **[ldap] ssl-keyfile** entry that is specified in each server's configuration file. If it is specified, then the **[ldap] ssl-keyfile** referenced in each `.conf` file on the same server as the `ldap.conf` containing the **[server:<instance>] ssl-keyfile-dn** value must contain a certificate with this label.

Default value

None.

Example

```
ssl-keyfile-dn = "PD_LDAP"
```

suffix

Use the **suffix** stanza entry to define federated registry suffixes.

Syntax

```
suffix = dn
```

Description

The **suffix** stanza entry defines the federated registry suffixes within the Security Verify Access registry that are stored on a back-end server instead. All operations on entries under these suffixes occur on the back-end server.

Mapping of the DNs is not possible. So the back-end server must contain a suffix of the same name.

The most specific (longest matching) suffix is selected. So overlapping suffixes can be defined.

Multiple entries are allowed.

Options

dn
Distinguished name of the federated registry LDAP suffix.

Usage

This stanza entry is optional.

Default value

None.

Example

```
suffix = o=ibm,c=us
```

user-objectclass

Use the **user-objectclass** stanza entry to define the object class names to set when you create a native user entry in LDAP.

Syntax

```
user-objectclass = object_class_list
```

Description

Contain a list of comma-separated object class names to set when you create a native user entry in LDAP.

Options

object_class_list
A list of comma-separated object class names.

Usage

This stanza entry is optional.

Default value

Default value is specific to the LDAP server that is detected.

Example

```
user-objectclass = top,person,organizationalPerson,inetOrgPerson,ePerson
```

user-search-filter

Use the **user-search-filter** entry to specify the LDAP search filter that is used by Security Verify Access.

Syntax

```
user-search-filter = ldap search filter
```

Description

Use this configuration file parameter to specify how to locate Security Verify Access users in LDAP.

Options

Specifies the LDAP search filter that is used by Security Verify Access to locate users in the LDAP directory server. This filter must be a valid LDAP string search filter as described by the Request for Comments (RFC) 2254 document.

Use the **user-search-filter** option with **user-objectclass** so that the Security Verify Access can locate LDAP users that are created with the LDAP object classes.

Do not update the unsupported option with the same name under the **[ldap-generic-general]** stanza.

Usage

This stanza entry is optional.

Default value

Default value is specific to the LDAP server that is detected.

Example

This example specifies a search for a `User` or `Person` under `user-objectclass`.

```
user-search-filter = (|(user-objectclass=User)(user-objectclass=Person))
```

[session-cookie-domains] stanza

domain

Use the **domain** stanza entry to specify domains that share the domain session cookie.

Syntax

```
domain = url
```

Description

Normally WebSEAL session cookies are host cookies that browsers only return to the host that originally set them.

This stanza is used to configure domain session cookies that are sent to any host in a particular DNS domain.

Options

url
Domains that share the domain cookie.

Usage

This stanza entry is optional.

Default value

None.

Example

```
domain = example.com
```

[session-http-headers] stanza

header_name

Syntax

```
header_name = {http|https}
```

Description

Configures HTTP headers to maintain session state.

Options

http

Configures HTTP headers to maintain session state over the HTTP transport.

https

Configures HTTP headers to maintain session state over the HTTPS transport.

Usage

This stanza entry is optional.

Default value

None.

Example

```
entrust-client = https
```

[snippet-filter] stanza

Use the **[snippet-filter]** stanza to configure parameters associated with the snippet filter.

max-snippet-size

The **max-snippet-size** stanza entry defines the maximum size (in bytes) of snippets that can be cached in memory.

Syntax

```
max-snippet-size = size_in_bytes
```

Description

If the snippet exceeds the configured maximum size, it cannot be cached. Instead, it will be read from the disk during the construction of each response.

Options

size_in_bytes

The maximum size (in bytes) of snippets that can be cached in memory.

Usage

This stanza entry is optional.

Default value

None.

Example

```
max-snippet-size = 1024
```

pattern-match-uri

The `pattern-match-uri` stanza entry defines whether wildcard characters are used when matching the snippet filter URI.

Syntax

```
pattern-match-uri = true|false
```

Description

If pattern matching is enabled each configured URI will be pattern-matched in order, and the snippet configuration for the first matching URI will be used.

Options

pattern-match-uri
Whether wildcard characters can be used to match a snippet filter URI.

Usage

This stanza entry is optional

Default value

`false`

Example

```
pattern-match-uri = false
```

[snippet-filter:*<uri>*] stanza

Use the **[snippet-filter:<uri>]** stanza to configure the snippet filter for a particular resource.

Syntax

```
[snippet-filter:<uri>]
<location> = <filename>
```

Description

This filter allows snippets to be inserted into the response for the specified URI. The filter examines each individual line of the response, and if the entire line matches, the filter inserts the snippet prior to the matching line.

Options

<uri>

The URI for which the snippet substitution takes place. The '*.' wildcard characters can be used when matching the URI, but only if the '`pattern-match-uri`' configuration entry in the `[snippet-filter]` stanza has been set to true. See “pattern-match-uri” on page 495.

<location>
The location at which the snippet is inserted. The location is pattern matched against a single line in the response. The snippet is then inserted immediately before the matching line. The value must represent the entire line that the snippet is to be inserted before. You can use the "*" and "?" wildcard characters in the location pattern. The pattern match for this parameter is case-sensitive.

<filename>
The name of the file which contains the snippet to be inserted. You can specify a path that is relative to the `snippet` directory in the management root directory. This parameter is case-sensitive.

Usage

You can specify multiple resources and configure multiple locations for each resource. The entries in the stanza must appear in the order in which they will be inserted in the returned page. Only the first match for each configuration entry is updated with the snippet. After a match has been found, WebSEAL progresses to the next configuration entry. If a match is not found for a particular configuration entry, the subsequent entries in the stanza are not processed.

You must include a new line character at the end of the snippet file contents if you want the snippet to be inserted as a separate line. Otherwise, WebSEAL adds the snippet to the start of the matching line.

When you configure a snippet-filter stanza for a resource that is accessed over a virtual junction or a local junction, the *<uri>* value is a forward slash followed by the resource name. For example, `[snippet-filter:/myResource.html]`.

As a result, a particular **`[snippet-filter:<uri>]`** stanza can represent multiple resources. In the preceding example, WebSEAL inserts snippets into the responses when accessing `myResource.html` over a local junction or virtual junction.

If the configured `<filename>` value is a non-existent file, WebSEAL fails to restart with a "DPWIV0752E Could not open file" error.

Default value

None.

Example

```
[snippet-filter:/stdjunction/sampleResource.html]
*test* = samplesnippet.html
?hello* = samplesnippet2.html
```

[spnego] stanza

Use the **`[spnego]`** stanza to configure Kerberos authentication.

spnego-auth

Use the **`spnego-auth`** stanza entry to enable Kerberos authentication.

Syntax

```
spnego-auth = {none|http|https|both}
```

Description

Enables authentication using the SPNEGO authentication mechanism.

Options

{none|http|https|both}
Specifies which protocols are supported. The value `both` means both HTTP and HTTPS.

Usage

This stanza entry is required.

Default value

none

Example

```
spnego-auth = none
```

spnego-krb-keytab-file

Use the **spnego-krb-keytab-file** stanza entry to specify the Kerberos keytab file.

Syntax

```
spnego-krb-keytab-file = keytab_file_name
```

Description

The name of the Kerberos keytab file for the WebSEAL server.

Options

keytab_file_name
The name of the Kerberos keytab file for the WebSEAL server.

Usage

This stanza entry is required.

Default value

None.

Example

```
spnego-krb-keytab-file = diamond_HTTP.keytab
```

spnego-krb-service-name

Use the **spnego-krb-service-name** stanza entry to specify the Kerberos service principal names for the server.

Syntax

```
spnego-krb-service-name = kerberos_server_principal_name
```

Description

Specifies the list of Kerberos service principal name (SPNs) for the server. Create a separate entry for each Kerberos service principal name.

Options

kerberos_server_principal_name

This name is created by combining the string HTTP with the hostname. The syntax is:

```
HTTP@host_name
```

The host name is the DNS name by which browsers contact the Web server. Use the fully qualified host name.

Usage

This stanza entry is required.

Default value

None.

Example

```
spnego-krb-service-name = HTTP@diamond.subnet2.ibm.com
spnego-krb-service-name = HTTP@ruby.subnet2.ibm.com
```

spnego-sid-attr-name

Use the **spnego-sid-attr-name** stanza entry to enable and define the credential attribute that stores the security identifier (SID) of the user.

Syntax

```
spnego-sid-attr-name = attribute_name
```

Description

During authentication, the system can add the SID of the user as an extended attribute to the credential. This stanza entry specifies the name of the attribute that stores the SID. The SID is extracted from the token that is used during Kerberos authentication.

Options

attribute_name
The name of the attribute to store the SID in.

Usage

This stanza entry is optional. If this entry is not present, then the system does not add the SID as an extended attribute to the credential.

Default value

None.

Example

```
spnego-sid-attr-name = kerberos_sid
```

use-domain-qualified-name

Use the **use-domain-qualified-name** stanza entry to control whether Security Verify Access includes the domain from the Kerberos user principal name as part of the Security Verify Access user ID.

Syntax

```
use-domain-qualified-name = {yes|no}
```

Description

Kerberos authentication provides a principal name of the form shortname@domain.com. By default, Security Verify Access uses only the short name as the Security Verify Access user ID. If this parameter is set to `yes`, then Security Verify Access includes the domain as part of the Security Verify Access user ID.

Note: The **use-domain-qualified-name** stanza entry has no effect if multiple-domain Active Directory is used as the Security Verify Access user registry. In this case, the domain name is always included as part of the Security Verify Access user name.

Options

yes

Security Verify Access includes the domain portion of the principal name as part of the Security Verify Access user ID. For example, say that Kerberos authentication provides a principal name of `user@example.com`. If **use-domain-qualified-name** is no, then the Security Verify Access user ID is `user`. If **use-domain-qualified-name** is `yes`, then the Security Verify Access name is `user@example.com`.

no

Security Verify Access uses only the short name as the Verify Access user ID, and does *not* include the domain portion of the principal name.

Usage

This stanza entry is required.

Default value

no

Example

```
use-domain-qualified-name = yes
```

spnego-ignore-ntlm-requests

Use the **spnego-ignore-ntlm-requests** stanza entry to ignore NTLM authentication requests during Kerberos authentication.

Syntax

```
spnego-ignore-ntlm-requests = {true|false}
```

Description

In some environments, clients can erroneously attempt NTLM authentication. By default, WebSEAL returns an error page that indicates that NTLM authentication is not supported. To instead ignore the NTLM authentication request and use the available configured authentication mechanisms, set this value to true.

Options

`true`
WebSEAL ignores NTLM authentication requests received from clients and attempts to authenticate by using configured authentication mechanisms.

false
WebSEAL returns an error page that indicates that NTLM is not supported when an NTLM authentication request is received.

Usage

This stanza entry is optional.

Default value

`false`

Example

```
spnego-ignore-ntlm-requests = true
```

[ssl] stanza

base-crypto-library

Use the **`base-crypto-library`** stanza entry to specify the cipher engine that GSKit uses.

Syntax

```
base-crypto-library = {Default|RSA|ICC}
```

Description

Specifies the cipher engine that is used by GSKit.

Options

`Default`
The value `Default` tells GSKit to use the optimal cryptographic base.

`RSA`
Use RSA.

Note: Setting this entry to RSA affects the settings possible for **fips-mode-processing**.

`ICC`
Use ICC.

Usage

This stanza entry is required.

Default value

`Default`

Example

```
base-crypto-library = Default
```

crl-ldap-server

Use the **`crl-ldap-server`** stanza entry in the **`[ssl]`** stanza to specify the LDAP server that WebSEAL can contact for CRL checking during client-side certificate authentication.

Syntax

```
crl-ldap-server = server_name
```

Description

Specifies the Server to be contacted to obtain Certificate Revocation Lists (CRL).

Options

server_name

This parameter can be set to one of two types of values:

1. The name of the LDAP server to be referenced as a source for Certificate Revocation Lists (CRL) during authentication across SSL junctions. If this is used, you may also need to set the following parameters:
   - crl-ldap-server-port
   - crl-ldap-user
   - crl-ldap-user-password
2. The literal string “URI”. In the case where no direct LDAP Server is available, this allows GSKit to obtain revocation information from LDAP or the HTTP Servers as specified by the CA in the CRL Distribution Point (CDP) extension of the certificate.

Usage

This stanza entry is optional.

Default value

None.

Example

```
crl-ldap-server = diamond.example.com
```

crl-ldap-server-port

Use the **crl-ldap-server-port** entry in the **[ssl]** stanza to set the port number for WebSEAL to use when it communicates with the LDAP server specified in **crl-ldap-server**.

Syntax

```
crl-ldap-server-port = port_number
```

Description

Port number for communication with the LDAP server specified in **crl-ldap-server**. The LDAP server is referenced for Certificate Revocation List (CRL) checking during SSL authentication.

Options

port_number
Port number for communication with the LDAP server specified in **crl-ldap-server**.

Usage

This stanza entry is optional. When **crl-ldap-server** is set, this stanza entry is required.

Default value

None.

Example

```
crl-ldap-server-port = 389
```

crl-ldap-user

Use the **crl-ldap-user** entry in the **[ssl]** stanza to specify an LDAP user who has permissions to retrieve the CRL on the LDAP server that is specified in **crl-ldap-server**.

Syntax

```
crl-ldap-user = user_DN
```

Description

Fully qualified distinguished name (DN) of an LDAP user that has access to the Certificate Revocation List.

Options

user_DN
Fully qualified distinguished name (DN) of an LDAP user that has access to the Certificate Revocation List.

Usage

This stanza entry is optional. A null value for **crl-ldap-user** indicates that the SSL authenticator should bind to the LDAP server anonymously.

Default value

None.

Example

```
crl-ldap-user =
cn=webseald/diamond,cn=SecurityDaemons,secAuthority=Default
```

crl-ldap-user-password

Use the **crl-ldap-user-password** entry in the **[ssl]** stanza to provide the password for the LDAP user that is specified in **crl-ldap-user**.

Syntax

```
crl-ldap-user-password = password
```

Description

Password for the user specified in **crl-ldap-user**.

Options

password
Password for the user specified in **crl-ldap-user**.

Usage

This stanza entry is optional.

Default value

None.

Example

```
crl-ldap-user-password = mypassw0rd
```

disable-ssl-v2

Use the **disable-ssl-v2** entry in the **[ssl]** stanza to control whether support for SSL version 2 is enabled in WebSEAL.

Syntax

```
disable-ssl-v2 = {yes|no}
```

Description

Disables support for SSL version 2. Support for SSL v2 is disabled by default. The WebSEAL configuration sets this value.

Options

yes
Support is disabled.

no
Support is enabled.

Usage

This stanza entry is optional. When not specified, the default is `yes`.

Default value

`yes`

Example

```
disable-ssl-v2 = yes
```

disable-ssl-v3

Use the **`disable-ssl-v3`** entry in the **`[ssl]`** stanza to control whether support for SSL version 3 is enabled in WebSEAL.

Syntax

```
disable-ssl-v3 = {yes|no}
```

Description

Disables support for SSL Version 3. Support for SSL V3 is enabled by default. The WebSEAL configuration sets this value.

Options

yes
The value `yes` means support is disabled.

no
The value no means the support is enabled.

Usage

This stanza entry is optional. When not specified, the default is no.

Default value

no

Example

```
disable-ssl-v3 = no
```

disable-tls-v1

Use the **`disable-tls-v1`** entry in the **`[ssl]`** stanza to control whether support for TLS version 1 is enabled in WebSEAL.

Syntax

```
disable-tls-v1 = {yes|no}
```

Description

Disables support for TLS Version 1. Support for TLS V1 is enabled by default. The WebSEAL configuration sets this value.

Options

yes
The value yes means support is disabled

no
The value no means the support is enabled.

Usage

This stanza entry is optional. When not specified, the default is no.

Default value

no

Example

```
disable-tls-v1 = no
```

disable-tls-v11

Use the **disable-tls-v11** entry in the **[ssl]** stanza to control whether support for TLS version 1.1 is enabled in WebSEAL.

Syntax

```
disable-tls-v11 = {yes|no}
```

Description

Determines whether WebSEAL supports Transport Layer Security (TLS) version 1.1. WebSEAL does not support TLS version 1.1 by default.

Options

yes
The value yes disables support for TLS version 1.1.

no
The value no enables support for TLS version 1.1.

Usage

This stanza entry is optional. If this entry is not specified, the default is yes.

Default value

yes

Example

```
disable-tls-v11 = yes
```

disable-tls-v12

Use the **disable-tls-v12** entry in the **[ssl]** stanza to control whether support for TLS version 1.2 is enabled in WebSEAL.

Syntax

```
disable-tls-v12 = {yes|no}
```

Description

Determines whether WebSEAL supports Transport Layer Security (TLS) version 1.2. WebSEAL supports TLS version 1.2 by default.

Options

yes
The value yes disables support for TLS version 1.2.

no
The value no enables support for TLS version 1.2.

Usage

This stanza entry is optional. If this entry is not specified, the default is no.

Default value

no

Example

```
disable-tls-v12 = no
```

disable-tls-v13

Use the disable-tls-v13 entry in the [ssl] stanza to control whether support for TLS version 1.3 is enabled in WebSEAL.

Syntax

```
disable-tls-v13 = {yes|no}
```

Description

Determines whether WebSEAL supports Transport Layer Security (TLS) version 1.3. Support for TLS version 1.3 is disabled by default.

Options

yes
Disables support for TLS version 1.3

no
Enables support for TLS version 1.3

Usage

This stanza entry is optional. If this entry is not specified, the default is yes.

disable-tls-v13=no disables DPWNS0302W messages.

Default value

yes

Example

```
disable-tls-v13 = no
```

enable-duplicate-ssl-dn-not-found-msgs

Use the **enable-duplicate-ssl-dn-not-found-msgs** stanza entry to control whether WebSEAL logs a warning whenever you connect to a junction that has the **-K** or **-B** flag set without the **-D** flag. WebSEAL can log duplicate messages every time it opens a connection to the junction or log a single warning only for each affected junction.

Syntax

```
enable-duplicate-ssl-dn-not-found-msgs = {yes | no}
```

Description

Determines whether WebSEAL logs a warning message every time you open a connection to a junction that has:

- Either the -K or the -B flag set, but
- The -D flag is not set.

By default, WebSEAL logs duplicate messages whenever it opens another connection to the junction. These messages appear in the following format:

```
DPWIV1212W   No server DN is defined for 'server.ibm.com'.
   The junctioned server DN verification is not performed."
```

Options

yes
Duplicate messages are created. Every time a connection is opened to a junction that has the -K or -B flags specified without the -D option, WebSEAL logs a warning.

no
When the server starts, WebSEAL logs a single warning only for each affected junction.

Usage

This stanza entry is required.

Default value

yes

Example

```
enable-duplicate-ssl-dn-not-found-msgs = no
```

fips-mode-processing

Use the **fips-mode-processing** stanza entry to enable or disable FIPS mode processing.

Syntax

```
fips-mode-processing = {yes|no}
```

Description

Enables or disables FIPS mode processing.

Options

yes
A value of yes enables FIPS mode processing.

no
A value of no disables FIPS mode processing. When `base-crypto-library = RSA`, this value must be no.

Usage

This stanza entry is required.

Default value

no

Example

```
fips-mode-processing = no
```

gsk-attr-name

Syntax

```
gsk-attr-name = {enum | string | number}:id:value
```

Description

Specify additional GSKit attributes to use when initializing an SSL connection with the client. A complete list of the available attributes is included in the GSKit SSL API documentation. This configuration entry can be specified multiple times. Configure a separate entry for each GSKit attribute.

Options

{enum | string | number}
The GSKit attribute type.

id
The identity associated with the GSKit attribute.

value
The value for the GSKit attribute.

Usage

This stanza entry is optional.

See “Appendix: Supported GSKit attributes” on page 583 for a list of GSKit attributes that can be configured.

You cannot configure the following restricted GSKit attributes:

```
GSK_BASE_CRYPTO_LIBRARY
GSK_SSL_FIPS_MODE_PROCESSING
GSK_FIPS_MODE_PROCESSING
GSK_OCSP_ENABLE
GSK_OCSP_URL
GSK_OCSP_NONCE_GENERATION_ENABLE
GSK_OCSP_NONCE_CHECK_ENABLE
GSK_OCSP_REQUEST_SIGKEYLABEL
GSK_OCSP_REQUEST_SIGALG
GSK_OCSP_PROXY_SERVER_NAME
GSK_OCSP_PROXY_SERVER_PORT
GSK_OCSP_RETRIEVE_VIA_GET
GSK_OCSP_MAX_RESPONSE_SIZE
GSK_KEYRING_FILE
GSK_KEYRING_PW
GSK_CRL_CACHE_SIZE
GSK_CRL_CACHE_ENTRY_LIFETIME
GSK_KEYRING_STASH_FILE
GSK_KEYRING_LABEL
GSK_LDAP_SERVER
GSK_LDAP_SERVER_PORT
GSK_LDAP_USER
GSK_LDAP_USER_PW
GSK_ACCELERATOR_NCIPHER_NF
GSK_ACCELERATOR_RAINBOW_CS
GSK_PKCS11_DRIVER_PATH
GSK_PKCS11_TOKEN_LABEL
GSK_PKCS11_TOKEN_PWD
GSK_PKCS11_ACCELERATOR_MODE
GSK_V2_SESSION_TIMEOUT
GSK_V3_SESSION_TIMEOUT
GSK_PROTOCOL_SSLV2
GSK_PROTOCOL_SSLV3
GSK_PROTOCOL_TLSV1
GSK_CLIENT_AUTH_TYPE
GSK_SESSION_TYPE
GSK_IO_CALLBACK
GSK_RESET_SESSION_TYPE_CALLBACK
GSK_RESET_SESSION_TYPE_CALLBACK
GSK_NO_RENEGOTIATION
GSK_ALLOW_ABBREVIATED_RENEGOTIATION
```

If you attempt to modify any of these attributes then an error message will be generated.

Default value

None.

Example

The following entry is for the GSKit attribute GSK_HTTP_PROXY_SERVER_NAME, which has an identity value of 225:

```
gsk-attr-name = string:225:proxy.ibm.com
```

See also

“gsk-attr-name” on page 78
“gsk-attr-name” on page 561
“jct-gsk-attr-name” on page 511

gsk-crl-cache-entry-lifetime

Syntax

```
gsk-crl-cache-entry-lifetime = number_of_seconds
```

Description

Integer value specifying the lifetime timeout, in seconds, for individual entries in the GSKit CRL cache.

See also the standards documents for SSL V3 and TLS V1 (RFC 2246) for more information on CRLs.

Options

number_of_seconds
Integer value specifying the lifetime timeout, in seconds, for individual entries in the GSKit CRL cache. The minimum value is 0. The maximum value is 86400. Neither WebSEAL nor GSKit impose a maximum value on the cache entry lifetime.

Usage

This stanza entry is required.

Default value

0

Example

```
gsk-crl-cache-entry-lifetime = 0
```

gsk-crl-cache-size

Syntax

```
gsk-crl-cache-size = number_of_entries
```

Description

Integer value indicating the maximum number of entries in the GSKit CRL cache.

See the standards documents for SSL V3 and TLS V1 (RFC 2246) for more information on CRLs.

Options

number_of_entries
Integer value indicating the maximum number of entries in the GSKit CRL cache. Minimum value is 0. A value of 0 means that no entries are cached. Neither WebSEAL nor GSKit impose a maximum value on this cache.

Usage

This stanza entry is required.

Default value

0

Example

```
gsk-crl-cache-size = 0
```

jct-gsk-attr-name

Syntax

```
jct-gsk-attr-name = {enum | string | number}:id:value
```

Description

Specify additional GSKit attributes to use when initializing an SSL connection with a junctioned server. A complete list of the available attributes is included in the GSKit SSL API documentation. This configuration entry can be specified multiple times. Configure a separate entry for each GSKit attribute.

This configuration item can be customized for a particular junction by adding the adjusted configuration item to a `[ssl:{jct_id}]` stanza, where *{jct-id}* refers to the junction point for a standard junction (include the leading /), or the virtual host label for a virtual host junction.

Options

{enum | string | number}
The GSKit attribute type.

id
The identity associated with the GSKit attribute.

value
The value for the GSKit attribute.

Usage

This stanza entry is optional.

See “Appendix: Supported GSKit attributes” on page 583 for a list of GSKit attributes that can be configured.

You cannot configure the following restricted GSKit attributes:

```
GSK_V2_SIDCACHE_SIZE
GSK_V3_SIDCACHE_SIZE
GSK_V2_SESSION_TIMEOUT
GSK_V3_SESSION_TIMEOUT
GSK_PROTOCOL_SSLV2
GSK_PROTOCOL_SSLV3
```

For a junction-specific **jct-gsk-attr-name** configuration entry, you cannot configure the following restricted GSKit attributes:

```
GSK_V2_SIDCACHE_SIZE
GSK_V3_SIDCACHE_SIZE
```

If you attempt to modify any of these attributes then an error message will be generated.

Default value

None.

Example

The following entry is for the GSKit attribute GSK_HTTP_PROXY_SERVER_NAME, which has an identity value of 225:

```
jct-gsk-attr-name = string:225:proxy.ibm.com
```

See also

“gsk-attr-name” on page 78
“gsk-attr-name” on page 508
“gsk-attr-name” on page 561

nist-compliance

Use the **nist-compliance** stanza entry to enable or disable NIST SP800-131A compliance.

Syntax

```
nist-compliance = {yes|no}
```

Description

Enables or disables NIST SP800-131A compliance.

Enabling NIST SP800-131A compliance results in the following automatic configuration:

- Enables FIPS mode processing.

  Note: When NIST SP800-131A compliance is enabled, FIPS mode processing is enabled regardless of the setting for the **fips-mode-processing** configuration entry.
- Enables TLS v1.2.

  Notes:

  – When NIST SP800-131A compliance is enabled, TLS v1.2 is enabled regardless of the setting for the **disable-tls-v12** configuration entry.
  – TLS v1 and TLS v1.1 are not disabled.
- Enables the appropriate signature algorithms.
- Sets the minimum RSA key size to 2048 bytes.

Options

yes
A value of yes enables NIST SP800-131A compliance.

no
A value of no disables NIST SP800-131A compliance.

Usage

This stanza entry is optional.

Default value

no

Example

```
nist-compliance = no
```

ocsp-enable

Syntax

```
ocsp-enable = {yes|no}
```

Description

Enable Online Certificate Status Protocol (OCSP) for checking the revocation status of certificates supplied by a server using the OCSP URL embedded in the certificate using an Authority Information Access (AIA) extension.

Options

yes
Enable OCSP to check the revocation status of server supplied certificates.

no
Disable OCSP checking of server supplied certificates.

Usage

This stanza entry is optional.

Note: This option can be used as an alternative to, or in conjunction with, the ocsp-url option.

Default value

no

Example

```
ocsp-enable = no
```

ocsp-max-response-size

Syntax

```
ocsp-max-response-size = number of bytes
```

Description

Sets the maximum response size (in bytes) that will be accepted as a response from an OCSP responder. This limit helps protect against a denial of service attack.

Options

number of bytes
Maximum response size, in bytes.

Note: A value of zero (0) indicates that the value is not set in the configuration file and no call to GSKit will be made to adjust its value; in this case, the option will assume the GSKit default of 20480 bytes.Non-zero values will be passed on to GSKit.

Usage

This stanza entry is optional.

Default value

204080

Example

```
ocsp-max-response-size = 20480
```

ocsp-nonce-check-enable

Syntax

```
ocsp-nonce-check-enable = {yes|no}
```

Description

Determines whether WebSEAL checks the nonce in the OCSP response. Enabling this option improves security but can cause OCSP Response validation to fail if there is a caching proxy between WebSEAL and the OCSP Responder. Note that enabling this option automatically enables the jct-ocsp-nonce-generation-enable option.

Options

yes
WebSEAL checks the nonce in the OCSP response to verify that it matches the nonce from the request.

no
WebSEAL does not check the nonce in the OCSP response.

Usage

This stanza entry is optional.

Default value

no

Example

```
ocsp-nonce-check-enable = no
```

ocsp-nonce-generation-enable

Syntax

```
ocsp-nonce-generation-enable = {yes|no}
```

Description

Determines whether WebSEAL generates a nonce as part of the OCSP request. Enabling this option can improve security by preventing replay attacks on WebSEAL but may cause an excessive load on an OCSP Responder appliance as the responder cannot use cached responses and must sign each response.

Options

yes
WebSEAL generates a nonce as part of the OCSP request.

no
WebSEAL does not generate a nonce as part of the OCSP request.

Usage

This stanza entry is optional.

Default value

no

Example

```
ocsp-nonce-generation-enable = no
```

ocsp-proxy-server-name

Syntax

```
ocsp-proxy-server-name = <proxy host name>
```

Description

Specifies the name of the proxy server that provides access to the OCSP responder.

Options

proxy host name
Fully qualified name of the proxy server.

Usage

This stanza entry is optional.

Default value

None

Example

```
ocsp-proxy-server-name = proxy.ibm.com
```

ocsp-proxy-server-port

Syntax

```
ocsp-proxy-server-port = <proxy host port number>
```

Description

Specifies the port number of the proxy server that provides access to the OCSP Responder.

Options

proxy host port number
Port number used by the proxy server to route OCSP requests and responses.

Usage

This stanza entry is optional.

Default value

None

Example

```
ocsp-proxy-server-port = 8888
```

ocsp-url

Syntax

```
ocsp-url = <OCSP Responder URL>
```

Description

Specifies the URL for the OCSP Responder. If a URL is provided, WebSEAL uses OCSP for all revocation status checking regardless of whether the certificate has an Authority Information Access (AIA) extension, which means that OCSP works with existing certificates. WebSEAL tries the OCSP Responder that is configured by this method first, rather than using a location specified by AIA extension. If revocation status is undetermined, and if **ocsp-enable** is set to yes, then WebSEAL tries to obtain revocation status using the access method in the AIA extension.

Options

OCSP Responder URL
URL of the OCSP Responder.

Usage

This stanza entry is optional.

Default value

None

Example

```
ocsp-url = http://responder.ibm.com/
```

pkcs11-keyfile

Use this entry to define the name of the pkcs11 key file that contains the configuration information for the network HSM device.

Syntax

```
pkcs11-keyfile = key_file
```

Description

If you have a network HSM device configured in your environment, use this entry to specify the name of the key file that contains the configuration for the device.

Options

key_file
The name of the pkcs11 key file that contains the configuration information for the network HSM device.

Usage

This stanza entry is optional.

Default value

None.

Example

```
pkcs11-keyfile = MyHSM
```

ssl-compliance

Specifies the SSL compliance mode.

Syntax

```
ssl-compliance = {fips|none|sp800-131-strict|sp800-131-transition|suite-b-128|suite-b-192}
```

Description

This stanza entry specifies the SSL compliance mode.

Note: The value of this stanza entry is set during the initial policy server configuration. Do not modify this value.

Options

fips

Enforces FIPS 140-2 protocols and algorithms. Security Verify Access servers and applications generate and use SHA1 with 2048-bit RSA certificates. Only TLS versions 1.0, 1.1, and 1.2 are available. SSL versions 2 and 3 are disabled and unavailable. This setting option is equivalent to the previous release setting `[ssl] ssl-enable-fips = yes`. This value is compatible with previous Tivoli Access Manager releases.

none

Specifies that no special compliance criteria are applied to TLS communication. Security Verify Access servers and applications generate and use SHA1 with 2048-bit RSA certificates. This setting option is equivalent to the previous release setting `[ssl] ssl-enable-fips = no`. This value is compatible with previous Tivoli Access Manager releases.

sp800-131-strict

Enables strict NIST SP800-131a support. This conformance enforcement is required by some agencies and businesses that start in the year 2014.

Security Verify Access servers and applications generate and use SHA256 with 2048-bit RSA certificates. This value is not compatible with prior releases of Tivoli Access Manager. Older Tivoli Access Manager clients cannot interact with Security Verify Access 7.0 running with this compliance setting. Only TLS version 1.2 is available; all others are disabled.

sp800-131-transition

Enables NIST SP800-131a support at the transition level. This value is valid until the end of the year 2013. This value has fewer restrictions than the strict enforcement. Only TLS versions 1.0, 1.1, and 1.2 are available. SSL versions 2 and 3 are disabled and unavailable.

Security Verify Access servers and applications generate and use SHA256 with 2048-bit RSA certificates. This value is at a higher level than is required by the standard and was chosen as it is a level that is permitted by the strict enforcement that allows easy migration from transition to strict. This value is not compatible with previous Tivoli Access Manager releases. Older Tivoli Access Manager clients cannot interact with Security Verify Access 7.0 running with this compliance setting.

suite-b-128

Enables NSA Suite B at 128-bit support. Security Verify Access servers and applications generate and use SHA256 with 256-bit ECDSA certificates. This value is not compatible with previous Tivoli Access Manager releases. Older Tivoli Access Manager clients cannot interact with Tivoli Access Manager 7.0 running with this compliance setting. Only TLS version 1.2 is available; all others are disabled.

suite-b-192

Enables NSA Suite B at 192-bit support. Security Verify Access servers and applications generate and use SHA384 with 384-bit ECDSA certificates. This value is not compatible with previous Tivoli Access Manager releases. Older Tivoli Access Manager clients cannot interact with Security Verify Access 7.0 running with this compliance setting. Only TLS version 1.2 is available; all others are disabled.

Usage

This stanza entry is optional.

Default value

None.

Example

```
ssl-compliance = fips
```

ssl-max-entries

Syntax

```
ssl-max-entries = number_of_entries
```

Description

Integer value indicating the maximum number of concurrent entries in the SSL cache.

Options

number_of_entries
Integer value indicating the maximum number of concurrent entries in the SSL cache. The minimum value is zero (0), which means that caching is unlimited. Entries between 0 and 256 are set to 256. There is no maximum limit.

Usage

This stanza entry is optional.

Default value

When the stanza entry is not assigned a value, WebSEAL uses a default value of 0.

Example

```
ssl-max-entries = 4096
```

ssl-v2-timeout

Syntax

```
ssl-v2-timeout = number_of_seconds
```

Description

Session timeout in seconds for SSL v2 connections between clients and servers. This timeout value controls how often a full SSL handshake is completed between clients and WebSEAL.

This value is set by the WebSEAL configuration utility.

Options

number_of_seconds
Valid range of values for *number_of_seconds* is from 1-100 seconds.

Usage

This stanza entry is required when SSL is enabled.

Default value

100

Example

```
ssl-v2-timeout = 100
```

ssl-v3-timeout

Syntax

```
ssl-v3-timeout = number_of_seconds
```

Description

Session timeout in seconds for SSL v3 connections between clients and servers. This timeout value controls how often a full SSL handshake is completed between clients and WebSEAL.

This value is set by the WebSEAL configuration utility.

Options

number_of_seconds
Valid range of values for *number_of_seconds* is from 10-86400 seconds, where 86400 seconds equals one day. If you specify a number outside this range, the server will generate an error and fail to start.

Usage

This stanza entry is required when SSL is enabled.

Default value

7200

Example

```
ssl-v3-timeout = 7200
```

suppress-client-ssl-errors

Syntax

```
suppress-client-ssl-errors = {true|false}
```

Description

This stanza entry suppresses error messages that originate from SSL communication problems with the client.

Options

true
Suppress error messages that originate from SSL communication problems with the client.

false
Do not suppress error messages that originate from SSL communication problems with the client.

Usage

This stanza entry is required when SSL is enabled.

Default value

false

Example

```
suppress-client-ssl-errors = false
```

undetermined-revocation-cert-action

Syntax

```
undetermined-revocation-cert-action = {ignore | log | reject}
```

Description

Controls the action that WebSEAL takes if OCSP or CRL is enabled but the responder cannot determine the revocation status of a certificate (that is, the revocation status is unknown). The appropriate values for this entry should be provided by the OCSP or CRL Responder owner.

Options

ignore
WebSEAL ignores the undetermined revocation status and permits use of the certificate.

log
WebSEAL logs the fact that the certificate status is undetermined and permits use of the certificate.

reject
WebSEAL logs the fact that the certificate status is undetermined and rejects the certificate.

Usage

This stanza entry is required.

Default value

The option defaults to `ignore` if it is not specified in the configuration file.

Note: The value for this option in the template configuration file is `log`.

Example

```
undetermined-revocation-cert-action = log
```

webseal-cert-keyfile

Syntax

```
webseal-cert-keyfile = file_name
```

Description

Specifies the WebSEAL certificate keyfile. This is the server certificate that WebSEAL exchanges with browsers when negotiating SSL sessions.

Options

file_name
Name of the WebSEAL certificate keyfile.

Usage

This stanza entry is required.

Default value

`pdsrv.kdb`

Example

```
webseal-cert-keyfile = pdsrv.kdb
```

webseal-cert-keyfile-label

Syntax

```
webseal-cert-keyfile-label = label_name
```

Description

String that specifies a label to use for the WebSEAL certificate keyfile. If this entry is not specified or an invalid label name is configured (that is, there is no certificate associated with the configured label), the label of the first certificate in the keystore is used. The keystore is specified by the **webseal-cert-keyfile** entry.

Options

label_name
String that specifies a label to use for the WebSEAL certificate keyfile.

Usage

This stanza entry is optional, but is set by default during WebSEAL configuration.

Default value

`WebSEAL-Test-Only`

Example

```
webseal-cert-keyfile-label = WebSEAL-Test-Only
```

webseal-cert-keyfile-sni

Use the **webseal-cert-keyfile-sni** stanza entry to configure WebSEAL to send a server certificate that contains a host name, which matches the host name in the initial browser request.

Syntax

```
webseal-cert-keyfile-sni = <host_name>:<label>
```

Description

This configuration has the following requirements:

- The user uses TLS over SSL to connect to WebSEAL. SSLv2 and SSLv3 are not supported.
- The browser supports Server Name Indication.

Use the **webseal-cert-keyfile-sni** configuration entry to specify the certificate that WebSEAL sends for a particular host name.

You can specify this configuration entry multiple times. Specify a separate entry for each server certificate.

If WebSEAL does not find an entry for the host name in the browser request, WebSEAL sends the default certificate that is specified by the **webseal-cert-keyfile-label** entry. WebSEAL also uses the default certificate if the request does not meet the Server Name Indication requirements. For example, if the browser does not support Server Name Indication.

Options

<host_name>
The name of the host to which WebSEAL returns the certificate.

<label>
The label of the certificate for WebSEAL to use.

Note: Specify the certificate that contains a **dn** value of cn=*<host_name>*.

Usage

This stanza entry is optional.

Default value

None.

Example

```
webseal-cert-keyfile-sni = hostA.abc.ibm.com:hostAcert
webseal-cert-keyfile-sni = vhostB.abc.ibm.com:vhostBcert
```

webseal-cert-keyfile-stash

Syntax

```
webseal-cert-keyfile-stash = file_name
```

Description

Name of the file containing an obfuscated version of the password used to protect private keys in the keyfile.

Options

file_name
Name of the file containing an obfuscated version of the password used to protect private keys in the keyfile.

Usage

This stanza entry is optional.

Default value

`pdsrv.sth`

Example

```
webseal-cert-keyfile-stash = pdsrv.sth
```

[ssl:<jct-id>] stanza

jct-gsk-attr-name

Syntax

```
jct-gsk-attr-name = {enum | string | number}:id:value
```

Description

Specify additional GSKit attributes to use when initializing an SSL connection with a junctioned server. A complete list of the available attributes is included in the GSKit SSL API documentation. This configuration entry can be specified multiple times. Configure a separate entry for each GSKit attribute.

This configuration item can be customized for a particular junction by adding the adjusted configuration item to a `[ssl:{jct_id}]` stanza, where *{jct-id}* refers to the junction point for a standard junction (include the leading /), or the virtual host label for a virtual host junction.

Options

{enum | string | number}
The GSKit attribute type.

id
The identity associated with the GSKit attribute.

value
The value for the GSKit attribute.

Usage

This stanza entry is optional.

See “Appendix: Supported GSKit attributes” on page 583 for a list of GSKit attributes that can be configured.

You cannot configure the following restricted GSKit attributes:

```
GSK_V2_SIDCACHE_SIZE
GSK_V3_SIDCACHE_SIZE
GSK_V2_SESSION_TIMEOUT
GSK_V3_SESSION_TIMEOUT
GSK_PROTOCOL_SSLV2
GSK_PROTOCOL_SSLV3
```

For a junction-specific **jct-gsk-attr-name** configuration entry, you cannot configure the following restricted GSKit attributes:

```
GSK_V2_SIDCACHE_SIZE
GSK_V3_SIDCACHE_SIZE
```

If you attempt to modify any of these attributes then an error message will be generated.

Default value

None.

Example

The following entry is for the GSKit attribute GSK_HTTP_PROXY_SERVER_NAME, which has an identity value of 225:

```
jct-gsk-attr-name = string:225:proxy.ibm.com
```

See also

“gsk-attr-name” on page 78
“gsk-attr-name” on page 508
“gsk-attr-name” on page 561

[ssl-qop] stanza

ssl-qop-mgmt

Syntax

```
ssl-qop-mgmt = {yes|no}
```

Description

Enables or disables SSL quality of protection management.

Options

yes
The value yes enables SSL quality of protection management.

no
The value no disables SSL quality of protection management.

Usage

This stanza entry is required.

Default value

no

Example

```
ssl-qop-mgmt = no
```

[ssl-qop-mgmt-default] stanza

default

Use the **default** entry in the **[ssl-qop-mgmt-default]** stanza to define the accepted encryption levels for access to WebSEAL over SSL.

Syntax

```
default = {ALL|NONE|cipher_level|cipher_name}
```

Description

List of string values to specify the allowed encryption levels for HTTPS access.

Values specified in this stanza entry are used for all IP addresses that are not matched in the **[ssl-qop-mgmt-networks]** stanza entries.

Note: The cipher suite must be set to TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 or higher to support HTTP/2 client connections.

Options

ALL
The value ALL allows all ciphers.

NONE
The value NONE disables all ciphers and uses an MD5 MAC check sum.

cipher_level
Legal cipher values are: NULL, DES-56, FIPS-DES-56, DES-168, FIPS-DES-168, RC2-40, RC2-128, RC4-40, RC4-56, RC4-128, AES-128, AES-256

Value	Cipher name in GSKit
NULL	TLS_RSA_WITH_NULL_MD5
DES-56	TLS_RSA_WITH_DES_CBC_SHA
FIPS-DES-56	SSL_RSA_FIPS_WITH_DES_CBC_SHA
DES-168	SSL_RSA_FIPS_WITH_3DES_EDE_CBC_SHA
FIPS-DES-168	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
RC2-40	TLS_RSA_EXPORT_WITH_RC2_CBC_40_MD5
RC2-128	TLS_RC2_CBC_128_CBC_WITH_MD5

Value	Cipher name in GSKit
RC4-40	TLS_RSA_EXPORT_WITH_RC4_40_MD5
RC4-56	TLS_RSA_EXPORT1024_WITH_RC4_56_SHA
RC4-128	TLS_RSA_WITH_RC4_128_MD5
AES-128	TLS_RSA_WITH_AES_128_CBC_SHA
AES-256	TLS_RSA_WITH_AES_256_CBC_SHA

cipher_name

Specific cipher names can also be used. This can be useful when the *cipher_level* above do not include a required cipher. When a cipher is enabled, it will be used with all enabled versions of SSL and TLS that support the cipher. The following is a list of available cipher names:

- SSL_RSA_FIPS_WITH_3DES_EDE_CBC_SHA
- SSL_RSA_FIPS_WITH_DES_CBC_SHA
- TLS_DHE_PSK_WITH_AES_128_CCM_8
- TLS_DHE_PSK_WITH_AES_128_CCM
- TLS_DHE_PSK_WITH_AES_256_CCM_8
- TLS_DHE_PSK_WITH_AES_256_CCM
- TLS_DHE_RSA_WITH_AES_128_CCM_8
- TLS_DHE_RSA_WITH_AES_128_CCM
- TLS_DHE_RSA_WITH_AES_128_GCM_SHA256
- TLS_DHE_RSA_WITH_AES_256_CCM_8
- TLS_DHE_RSA_WITH_AES_256_CCM
- TLS_DHE_RSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA
- TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_ECDSA_WITH_RC4_128_SHA
- TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_RSA_WITH_RC4_128_SHA
- TLS_PSK_WITH_AES_128_CCM_8
- TLS_PSK_WITH_AES_128_CCM
- TLS_PSK_WITH_AES_256_CCM_8

- TLS_PSK_WITH_AES_256_CCM
- TLS_RSA_EXPORT1024_WITH_DES_CBC_SHA
- TLS_RSA_EXPORT1024_WITH_RC4_56_SHA
- TLS_RSA_EXPORT_WITH_RC2_CBC_40_MD5
- TLS_RSA_EXPORT_WITH_RC4_40_MD5
- TLS_RSA_WITH_3DES_EDE_CBC_SHA
- TLS_RSA_WITH_AES_128_CBC_SHA256
- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_AES_128_CCM_8
- TLS_RSA_WITH_AES_128_CCM
- TLS_RSA_WITH_AES_128_GCM_SHA256
- TLS_RSA_WITH_AES_256_CBC_SHA256
- TLS_RSA_WITH_AES_256_CBC_SHA
- TLS_RSA_WITH_AES_256_CCM_8
- TLS_RSA_WITH_AES_256_CCM
- TLS_RSA_WITH_AES_256_GCM_SHA384
- TLS_RSA_WITH_DES_CBC_SHA
- TLS_RSA_WITH_NULL_MD5
- TLS_RSA_WITH_NULL_NULL
- TLS_RSA_WITH_NULL_SHA
- TLS_RSA_WITH_RC4_128_MD5
- TLS_RSA_WITH_RC4_128_SHA
- TLS_RSA_WITH_NULL_SHA256
- SSL_CK_RC4_128_WITH_MD5
- SSL_CK_RC4_128_EXPORT40_WITH_MD5
- SSL_CK_RC2_128_CBC_WITH_MD5
- SSL_CK_RC2_128_CBC_EXPORT40_WITH_MD5
- SSL_CK_DES_64_CBC_WITH_MD5
- SSL_CK_DES_192_EDE3_CBC_WITH_MD5
- TLS_ECDHE_ECDSA_WITH_NULL_SHA
- TLS_ECDHE_RSA_WITH_NULL_SHA
- TLS_AES_128_GCM_SHA256
- TLS_AES_256_GCM_SHA384
- TLS_CHACHA20_POLY1305_SHA256
- TLS_AES_128_CCM_SHA256
- TLS_AES_128_CCM_8_SHA256

Usage

This stanza entry is required.

Default value

```
# AES-128
default = TLS_AES_128_GCM_SHA256
default = TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
default = TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
```

```
default = TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
default = TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256

# AES-256
default = TLS_AES_256_GCM_SHA384
default = TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
default = TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
default = TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
default = TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
```

Note:

The following aliases are default values prior to version 10.0.0.0:

```
default = AES-128
default = AES-256
```

The legacy cipher aliases AES-128 and AES-256 are equivalent to the following default values:

```
# AES-128
default = TLS_AES_128_CCM_8_SHA256
default = TLS_AES_128_CCM_SHA256
default = TLS_AES_128_GCM_SHA256
default = TLS_RSA_WITH_AES_128_CBC_SHA
default = TLS_RSA_WITH_AES_128_CBC_SHA256
default = TLS_RSA_WITH_AES_128_GCM_SHA256

# AES-256
default = TLS_AES_256_GCM_SHA384
default = TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
default = TLS_RSA_WITH_AES_256_CBC_SHA256
default = TLS_RSA_WITH_AES_256_GCM_SHA384
```

Example

To specify a selected group of ciphers, create a separate entry for each cipher. For example:

```
default = RC4-128
default = RC2-128
default = DES-168
```

The following cipher is the minimum requirement for HTTP/2 over TLS and is not in the set of ciphers specified by the cipher alias "AES-128".

```
default = TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
default = AES-128
default = AES-256
```

[ssl-qop-mgmt-hosts] stanza

host-ip

Syntax

```
host-ip = {ALL|NONE|cipher_level|cipher_name}
```

Description

List of string values to specify the allowed encryption levels for HTTPS access for a specific IP address.

Note that this stanza has been deprecated and is retained only for backward compatibility.

Options

ALL
The value ALL allows all ciphers.

NONE
The value NONE disables all ciphers and uses an MD5 MAC check sum.

cipher_level
Legal cipher values are: NULL, DES-56, FIPS-DES-56, DES-168, FIPS-DES-168, RC2-40, RC2-128, RC4-40, RC4-56, RC4-128, AES-128, AES-256

Value	Cipher name in GSKit
NULL	TLS_RSA_WITH_NULL_MD5
DES-56	TLS_RSA_WITH_DES_CBC_SHA
FIPS-DES-56	SSL_RSA_FIPS_WITH_DES_CBC_SHA
DES-168	SSL_RSA_FIPS_WITH_3DES_EDE_CBC_SHA
FIPS-DES-168	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
RC2-40	TLS_RSA_EXPORT_WITH_RC2_CBC_40_MD5
RC2-128	TLS_RC2_CBC_128_CBC_WITH_MD5
RC4-40	TLS_RSA_EXPORT_WITH_RC4_40_MD5
RC4-56	TLS_RSA_EXPORT1024_WITH_RC4_56_SHA
RC4-128	TLS_RSA_WITH_RC4_128_MD5
AES-128	TLS_RSA_WITH_AES_128_CBC_SHA
AES-256	TLS_RSA_WITH_AES_256_CBC_SHA

cipher_name

Specific cipher names can also be used. This can be useful when the *cipher_level* above do not include a required cipher. When a cipher is enabled, it will be used with all enabled versions of SSL and TLS that support the cipher. The following is a list of available cipher names:

- SSL_RSA_FIPS_WITH_3DES_EDE_CBC_SHA
- SSL_RSA_FIPS_WITH_DES_CBC_SHA
- TLS_DHE_PSK_WITH_AES_128_CCM_8
- TLS_DHE_PSK_WITH_AES_128_CCM
- TLS_DHE_PSK_WITH_AES_256_CCM_8
- TLS_DHE_PSK_WITH_AES_256_CCM
- TLS_DHE_RSA_WITH_AES_128_CCM_8
- TLS_DHE_RSA_WITH_AES_128_CCM
- TLS_DHE_RSA_WITH_AES_128_GCM_SHA256
- TLS_DHE_RSA_WITH_AES_256_CCM_8
- TLS_DHE_RSA_WITH_AES_256_CCM
- TLS_DHE_RSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA
- TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8

- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_ECDSA_WITH_RC4_128_SHA
- TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_RSA_WITH_RC4_128_SHA
- TLS_PSK_WITH_AES_128_CCM_8
- TLS_PSK_WITH_AES_128_CCM
- TLS_PSK_WITH_AES_256_CCM_8
- TLS_PSK_WITH_AES_256_CCM
- TLS_RSA_EXPORT1024_WITH_DES_CBC_SHA
- TLS_RSA_EXPORT1024_WITH_RC4_56_SHA
- TLS_RSA_EXPORT_WITH_RC2_CBC_40_MD5
- TLS_RSA_EXPORT_WITH_RC4_40_MD5
- TLS_RSA_WITH_3DES_EDE_CBC_SHA
- TLS_RSA_WITH_AES_128_CBC_SHA256
- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_AES_128_CCM_8
- TLS_RSA_WITH_AES_128_CCM
- TLS_RSA_WITH_AES_128_GCM_SHA256
- TLS_RSA_WITH_AES_256_CBC_SHA256
- TLS_RSA_WITH_AES_256_CBC_SHA
- TLS_RSA_WITH_AES_256_CCM_8
- TLS_RSA_WITH_AES_256_CCM
- TLS_RSA_WITH_AES_256_GCM_SHA384
- TLS_RSA_WITH_DES_CBC_SHA
- TLS_RSA_WITH_NULL_MD5
- TLS_RSA_WITH_NULL_NULL
- TLS_RSA_WITH_NULL_SHA
- TLS_RSA_WITH_RC4_128_MD5
- TLS_RSA_WITH_RC4_128_SHA
- TLS_RSA_WITH_NULL_SHA256
- SSL_CK_RC4_128_WITH_MD5
- SSL_CK_RC4_128_EXPORT40_WITH_MD5
- SSL_CK_RC2_128_CBC_WITH_MD5
- SSL_CK_RC2_128_CBC_EXPORT40_WITH_MD5

- SSL_CK_DES_64_CBC_WITH_MD5
- SSL_CK_DES_192_EDE3_CBC_WITH_MD5
- TLS_ECDHE_ECDSA_WITH_NULL_SHA
- TLS_ECDHE_RSA_WITH_NULL_SHA
- TLS_AES_128_GCM_SHA256
- TLS_AES_256_GCM_SHA384
- TLS_CHACHA20_POLY1305_SHA256
- TLS_AES_128_CCM_SHA256
- TLS_AES_128_CCM_8_SHA256

Usage

This stanza entry is optional.

Default value

None.

Example

To specify allowable ciphers for a selected group of IP addresses, create a separate entry for each address. For example:

```
111.222.333.444 = RC4-128
222.666.333.111 = RC2-128
```

[ssl-qop-mgmt-networks] stanza

network/netmask

Syntax

```
network/netmask = {ALL|NONE|cipher_level|cipher_name}
```

Description

List of string values to specify the allowed encryption levels for HTTPS access for a specific combination of IP address and netmask.

Note that this stanza has been deprecated and is retained only for backward compatibility.

Options

ALL
The value ALL allows all ciphers.

NONE
The value NONE disables all ciphers and uses an MD5 MAC check sum.

cipher_level
Legal cipher values are: NULL, DES-56, FIPS-DES-56, DES-168, FIPS-DES-168, RC2-40, RC2-128, RC4-40, RC4-56, RC4-128, AES-128, AES-256

Value	Cipher name in GSKit
NULL	TLS_RSA_WITH_NULL_MD5
DES-56	TLS_RSA_WITH_DES_CBC_SHA
FIPS-DES-56	SSL_RSA_FIPS_WITH_DES_CBC_SHA
DES-168	SSL_RSA_FIPS_WITH_3DES_EDE_CBC_SHA
FIPS-DES-168	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
RC2-40	TLS_RSA_EXPORT_WITH_RC2_CBC_40_MD5
RC2-128	TLS_RC2_CBC_128_CBC_WITH_MD5
RC4-40	TLS_RSA_EXPORT_WITH_RC4_40_MD5
RC4-56	TLS_RSA_EXPORT1024_WITH_RC4_56_SHA
RC4-128	TLS_RSA_WITH_RC4_128_MD5
AES-128	TLS_RSA_WITH_AES_128_CBC_SHA
AES-256	TLS_RSA_WITH_AES_256_CBC_SHA

cipher_name

Specific cipher names can also be used. This can be useful when the *cipher_level* above do not include a required cipher. When a cipher is enabled, it will be used with all enabled versions of SSL and TLS that support the cipher. The following is a list of available cipher names:

- SSL_RSA_FIPS_WITH_3DES_EDE_CBC_SHA
- SSL_RSA_FIPS_WITH_DES_CBC_SHA
- TLS_DHE_PSK_WITH_AES_128_CCM_8
- TLS_DHE_PSK_WITH_AES_128_CCM
- TLS_DHE_PSK_WITH_AES_256_CCM_8
- TLS_DHE_PSK_WITH_AES_256_CCM
- TLS_DHE_RSA_WITH_AES_128_CCM_8
- TLS_DHE_RSA_WITH_AES_128_CCM
- TLS_DHE_RSA_WITH_AES_128_GCM_SHA256
- TLS_DHE_RSA_WITH_AES_256_CCM_8
- TLS_DHE_RSA_WITH_AES_256_CCM
- TLS_DHE_RSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA
- TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_ECDSA_WITH_RC4_128_SHA
- TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256

- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_RSA_WITH_RC4_128_SHA
- TLS_PSK_WITH_AES_128_CCM_8
- TLS_PSK_WITH_AES_128_CCM
- TLS_PSK_WITH_AES_256_CCM_8
- TLS_PSK_WITH_AES_256_CCM
- TLS_RSA_EXPORT1024_WITH_DES_CBC_SHA
- TLS_RSA_EXPORT1024_WITH_RC4_56_SHA
- TLS_RSA_EXPORT_WITH_RC2_CBC_40_MD5
- TLS_RSA_EXPORT_WITH_RC4_40_MD5
- TLS_RSA_WITH_3DES_EDE_CBC_SHA
- TLS_RSA_WITH_AES_128_CBC_SHA256
- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_AES_128_CCM_8
- TLS_RSA_WITH_AES_128_CCM
- TLS_RSA_WITH_AES_128_GCM_SHA256
- TLS_RSA_WITH_AES_256_CBC_SHA256
- TLS_RSA_WITH_AES_256_CBC_SHA
- TLS_RSA_WITH_AES_256_CCM_8
- TLS_RSA_WITH_AES_256_CCM
- TLS_RSA_WITH_AES_256_GCM_SHA384
- TLS_RSA_WITH_DES_CBC_SHA
- TLS_RSA_WITH_NULL_MD5
- TLS_RSA_WITH_NULL_NULL
- TLS_RSA_WITH_NULL_SHA
- TLS_RSA_WITH_RC4_128_MD5
- TLS_RSA_WITH_RC4_128_SHA
- TLS_RSA_WITH_NULL_SHA256
- SSL_CK_RC4_128_WITH_MD5
- SSL_CK_RC4_128_EXPORT40_WITH_MD5
- SSL_CK_RC2_128_CBC_WITH_MD5
- SSL_CK_RC2_128_CBC_EXPORT40_WITH_MD5
- SSL_CK_DES_64_CBC_WITH_MD5
- SSL_CK_DES_192_EDE3_CBC_WITH_MD5
- TLS_ECDHE_ECDSA_WITH_NULL_SHA
- TLS_ECDHE_RSA_WITH_NULL_SHA
- TLS_AES_128_GCM_SHA256
- TLS_AES_256_GCM_SHA384
- TLS_CHACHA20_POLY1305_SHA256

- TLS_AES_128_CCM_SHA256
- TLS_AES_128_CCM_8_SHA256

Usage

This stanza entry is optional.

Default value

None.

Example

To specify allowable ciphers for a selected group of IP addresses and netmasks, create a separate entry for each address/netmask combination. For example:

```
111.222.333.444/255.255.255.0 = RC4-128
222.666.333.111/255.255.0.0 = RC2-128
```

[sso:<service-name>] stanza

Use the `[sso:<service-name>]` stanza to configure a SSO service which will be used by the Web Reverse Proxy to manage SSO usernames and passwords.

sso-endpoint

Use the single sign-on endpoint to specify an endpoint that is called to manage SSO usernames and passwords.

Syntax

```
sso-endpoint = endpoint
```

Description

The endpoint from where the credentials can be retrieved. This should be a URL containing the macros `{user}` and `{resource}`.

Options

endpoint
Specifies the endpoint that is called to manage SSO usernames and passwords.

Usage

This stanza entry is required.

Default Value

None.

Example

```
sso-endpoint = https://vault.verify.ibm.com:9443/sso/{user}/resource/{resource}
```

proxy

Use this stanza entry to define the proxy, if any, which is used to reach the various endpoints.

Syntax

```
proxy = http[s]://<address>:<port>
```

Description

The proxy that is used to reach the various endpoints.

Options

<address>:<port>
The host name and port number of the proxy.

Usage

This stanza entry is optional.

Default Value

None.

Example

```
proxy = https://www.example.com:8080
```

user-id-attribute

The name of the credential attribute which is used to populate the `{user}` macro.

Syntax

```
user-id-attribute = attribute
```

Description

The name of the credential attribute which is used to populate the `{user}` macro in the `sso-endpoint`.

Options

attribute
The name of the credential attribute.

Usage

This stanza entry is required.

Default Value

None.

Example

```
user-id-attribute = AZN_CRED_PRINCIPAL_NAME
```

user-id-encoding

The encoding which is used for the user identity.

Syntax

```
user-id-encoding = encoding
```

Description

Controls how the `{user}` macro value is encoded in the URL used when communicating with the credential service.

Options

encoding
The type of encoding, either `url` or `base64url`. If the encoding is `base64url`, the reverse proxy will indicate this to the credential service by appending a query string parameter to the URL: `encoding=base64url`.

Usage

This stanza entry is required.

Default Value

None.

Example

```
user-id-encoding = url
```

encryption-key-label

The label of the key which is used to protect the credential data.

Syntax

```
encryption-key-label = label
```

Description

The label of the key from the Web Reverse Proxy key file which will be used to encrypt and decrypt credential data. This key should either be an RSA or ECDSA key for which the private and public parts are available.

Options

label
The label of the key from the Web Reverse Proxy key file.

Usage

This stanza entry is optional.

Default Value

None.

Example

```
encryption-key-label = sso-key
```

authentication-endpoint

The endpoint which will be called to obtain an access token for the SSO service.

Syntax

```
authentication-endpoint = endpoint
```

Description

This is the endpoint which will be called to obtain an access token which is used in requests to the credential service. If no endpoint is specified, a BA header constructed from the client-id and client-secret configuration entries will be used. The endpoint should conform to the OAuth client credential flow (OAuth 2.0 RFC 6749, section 4.4).

Options

endpoint
Specifies the endpoint that is called to obtain an access token.

Usage

This stanza entry is optional.

Default Value

None.

Example

```
sso-endpoint = https://vault.verify.ibm.com:9443/access_token
```

authentication-endpoint-payload

Defines how the client ID and secret will be provided to the authentication endpoint.

Syntax

```
authentication-endpoint-payload = payload-type
```

Description

Specifies how the client id and secret will be sent to the authentication endpoint. If `form` is specified, the client id and secret will be included in forms POST data. If `basic` is specified a basic authentication header will be constructed using the client id and secret.

Options

payload-type
The type of payload, either `form` or `basic`.

Usage

This stanza entry is optional.

Default Value

None.

Example

```
authentication-endpoint-payload = form
```

client-id

The client identifier, which is used when calling out to the authentication service.

Syntax

```
client-id = client-id
```

Description

The client identifier, which is used when authenticating to the callout services:

- If this entry is specified and authentication-endpoint and client-secret are not provided, this value will be used as a bearer token.
- If this value is surrounded by "{" and "}", the value will be substituted for the value of a credential attribute with the same name and used as a bearer token.
- If this entry is specified and authentication-endpoint is not provided, this value and client-secret will be used to construct a basic authentication header, where client-id is the username and client-secret is the password.

Options

client-id
The client identifier.

Usage

This stanza entry is optional.

Default Value

None.

Example

```
client-id = testuser
```

client-secret

The client secret, which is used when calling out to the authentication service.

Syntax

```
client-secret = client-secret
```

Description

The client secret, which is used when authenticating to the callout services. If this entry is specified and authentication-endpoint is not provided, this value and client-id will be used to construct a basic authentication header, where client-id is the username and client-secret is the password.

Options

client-secret
The client secret.

Usage

This stanza entry is optional.

Default Value

None.

Example

```
client-secret = secret
```

ssl-keyfile-label

The label of the certificate used to mutually authenticate to the SSO service.

Syntax

```
ssl-keyfile-label = label
```

Description

The label associated with the client key which is used to perform mutual authentication with the SSO service. This key must exist in the key file which is used by the Web Reverse Proxy.

Options

label
The label of the personal certificate to be used for mutual authentication.

Usage

This stanza entry is optional.

Default Value

None.

Example

```
ssl-keyfile-label = my-cert
```

ssl-valid-server-dn

The accepted DN of the certificate presented by the credential service.

Syntax

```
ssl-valid-server-dn = dn
```

Description

This entry specifies the accepted DN of the certificate presented by the credential service. If this entry is empty, any DN will be accepted.

Options

dn
The certificate DN.

Usage

This stanza entry is optional.

Default Value

None.

Example

```
ssl-valid-server-dn = cn=test,o=ibm
```

ssl-keyfile-sni

The server name (SNI) which will be indicated when establishing a connection to the credential service.

Syntax

```
ssl-keyfile-sni = sni
```

Description

The server name (SNI) which will be indicated when establishing a connection to the credential service. If this entry is empty, no name will be indicated.

Options

sni
The server name indicator.

Usage

This stanza entry is optional.

Default Value

None.

Example

```
ssl-keyfile-sni = www.ibm.com
```

[statistics] stanza

Use the `[statistics]` stanza to define the statistic collection and publishing capability for WebSEAL. The statistics for the configured components are published to the specified `statsd` server over UDP. Statistics can be used to monitor the environment or assist with problem determination in the environment.

component

Use the `component` stanza entry to specify the components for which statistics will be sent to the `statsd` server.

Syntax

```
component = <component>
```

Description

The name of a statistical component which is to be enabled. Statistics from enabled components are sent to the `statsd` server. For a list of available components, see Sending statistics to `Statsd`. This entry may be repeated multiple times, once for each component which is to be enabled.

Options

`component`
The name of the component which is to be enabled.

Usage

This stanza entry is required.

Default value

None

Example

```
component=pdweb.http
component=pdweb.authn
```

frequency

Use the frequency stanza entry to specify the frequency at which statistics are sent to the `statsd` server.

Syntax

```
frequency = <frequency>
```

Description

The frequency (in seconds) that statistics are sent from the memory buffer to the `statsd` server. The minimum value for this configuration entry is 1, indicating that statistics are sent every second.

Options

`frequency`
The frequency at which statistics are sent to the `statsd` server.

Usage

This stanza entry is required.

Default value

30

Example

```
frequency = 30
```

port

Use the `port` stanza entry to specify the port on which the remote `statsd` server is listening.

Syntax

```
port = <port>
```

Description

Specifies the port on which the `statsd` server is listening for requests.

Options

`port`
The port on which the `statsd` server is listening.

Usage

This stanza entry is optional.

Default value

8125

Example

```
port = 8125
```

prefix

Use the `prefix` stanza entry to specify the prefix which will be prepended to the statistic component name when generating the name of the metric.

Syntax

```
prefix = <prefix>
```

Description

Specifies the prefix which will be prepended to the statistic component name when generating the name of the metric which is sent to the `statsd` server.

Options

`prefix`
The prefix used when generating the metric name.

Usage

This stanza entry is optional.

Default value

None.

Example

```
prefix = instanceA.
```

server

Use the `server` stanza entry to specify the name or IP address of the remote `statsd` server.

Syntax

```
server = <server-name>
```

Description

Specifies the server name or IP address on which the `statsd` server is listening for requests. The generation of statistics only occurs if this entry is set.

Options

`server-name`
The name or IP address of the `statsd` server.

Usage

This stanza entry is required.

Default value

None.

Example

```
server = statsd.ibm.com
```

[step-up] stanza

retain-stepup-session

Syntax

```
retain-stepup-session = {yes|no}
```

Description

Determines whether a session cookie issued during a step-up operation is allowed to be reused or not. This option is only in effect if the **verify-step-up-user** option is set to yes.

Options

yes
Enables session cookie to be reused during a step-up operation.

no
Prevents session cookie from being reused during a step-up operation.

Usage

This stanza entry is required.

Default value

no

Example

```
retain-stepup-session = no
```

show-all-auth-prompts

Syntax

```
show-all-auth-prompts = {yes|no}
```

Description

Controls login prompt response for an unauthenticated user who requests an object protected by a step-up authentication POP attribute.

Options

yes
A value of "yes" provides multiple login prompts—one for each enabled authentication method—on each login page.

no
A value of "no" provides only the login prompt for the specific authentication level required by the POP(default).

Usage

This stanza entry is required.

Default value

no

Example

```
show-all-auth-prompts = no
```

step-up-at-higher-level

Syntax

```
step-up-at-higher-level = {yes|no}
```

Description

This configuration entry controls whether an authentication mechanism that is higher than the requested step-up level is accepted during a step-up operation.

Options

yes
Authentication levels higher than the level specified in the POP are accepted during step-up operations.

no
Higher authentication levels are disallowed during step-up operations.

Usage

This stanza entry is optional.

Default value

no

Example

```
step-up-at-higher-level = no
```

verify-step-up-user

Syntax

```
verify-step-up-user = {yes|no}
```

Description

Determines whether the identity of the user performing a step-up operation must match the identity of the user that performed the previous authentication.

Options

yes
The identity of the user performing the step-up operation must match the identity of the user that performed the previous authentication. In this case, the existing session key will be retained during step-up authentication. The value of the **retain-stepup-session** option controls whether the existing session key will be retained during step-up authentication.

no
The identity of the user performing the step-up operation need not match the identity of the user that performed the previous authentication operation. In this case, the session key must change during step-up authentication.

Usage

This stanza entry is required.

Default value

yes

Example

```
verify-step-up-user = yes
```

[system-environment-variables] stanza

env-name

Use the *env-name* stanza entry to list the system environment variables that the WebSEAL daemon exports during initialization.

Syntax

```
env-name = env-value
```

Description

Defines system environment variables that are exported by WebSEAL.

During initialization, the WebSEAL daemon exports the environment variables that are defined as entries in the **[system-environment-variables]** stanza. You must include a separate entry for each system environment variable that you want to export.

Options

env-name
The name of the system environment variable.

env-value
The value of the system environment variable.

Usage

This stanza entry is optional.

Notes:

- This functionality is not supported on Windows platforms.
- The environment variable names are case-sensitive.
- The PD_SVC_ROUTING_FILE environment variable is not supported.

Default value

None.

Example

The following example sets the LANG and GSK_TRACE_FILE environment variables.

```
LANG = de
GSK_TRACE_FILE = /tmp/gsk.trace
```

[tfimsso] stanza

Use this stanza to house the default TFIM single sign-on configuration information. This stanza will be used if no junction specific tfimsso stanza is provided.

always-send-tokens

Use the **always-send-tokens** stanza entry to control whether WebSEAL sends a security token for every HTTP request.

Syntax

```
always-send-tokens = {yes|true|no|false}
```

Description

Indicates whether WebSEAL sends a security token for every HTTP request or whether WebSEAL waits for a 401 response before it adds the security token.

You can use this configuration item to avoid generating and adding a security token to every request if the back-end web server can maintain user sessions. This configuration item is only useful if the request for authentication involves a 401 response, which currently applies to the Federation Runtime single sign-on only.

Options

yes
WebSEAL sends a security token for every HTTP request.

no
WebSEAL waits for a 401 response before it sends a security token for an HTTP request.

Usage

The `always-send-token` stanza entry is required when Federation SSO authentication is used over junctions.

Default value

None

Example

```
always-send-tokens = false
```

applies-to

Use the **`applies-to`** stanza entry to specify the location of the Security Token Service module if you are using Federation Runtime SSO authentication.

Syntax

```
applies-to = http://<webseal-server>/<junction>
```

Description

Path to specify the location to search for the appropriate Security Token Service (STS) module in Federation Runtime.

Options

http://<webseal-server>/<junction>
The host name or IP address of the WebSEAL server, along with the junction name. This address is similar to the URL that is used to access the junction.

Usage

The `applies-to` stanza entry is required when the Federation SSO authentication is used over junctions.

Default value

None

Example

```
applies-to = http://webseal-server/jct
```

one-time-token

Syntax

```
one-time-token = {true | false}
```

Description

This boolean value is used to indicate whether the security token that is produced by TFIM is only valid for a single transaction. An example of a one-time-token is a Kerberos token, which can only be used for a single authentication operation.

Usage

This stanza entry is required when TFIM SSO authentication is used over junctions.

Default value

True.

Example

```
one-time-token = false
```

preserve-xml-token

Syntax

```
preserve-xml-token = {true | false}
```

Description

This value controls whether to use the requested `BinarySecurityToken` XML structure in its entirety or whether only the encapsulated token should be used. Set this configuration entry to `true` only if the junctioned Web server understands and expects the `BinarySecurityToken` XML structure.

Usage

This stanza entry is required when TFIM SSO authentication is used over junctions.

Default value

True.

Example

```
preserve-xml-token = false
```

renewal-window

Syntax

```
renewal-window = number of seconds
```

Description

The length of time, in seconds, by which the expiration of security tokens will be reduced. This entry is used to make allowances for differences in system times and transmission times for the security tokens.

Options

number of seconds
Number of seconds by which the expiration of security tokens will be reduced to make allowances for differences between system times and transmission times for security tokens.

Usage

This stanza entry is required when TFIM SSO authentication is used over junctions.

Default value

None

Example

```
renewal-window = 15
```

service-name

Syntax

```
service-name = <servicename>
```

Description

1. Used by TFIM when searching for a matching trust chain. This configuration entry will be compared against the configured **AppliesTo** service name value for each trust chain. The second field within the **AppliesTo** service name configuration entry should be set to either asterisk (*) to match all service names, or it should be set to the value defined by this configuration item. See the TFIM documentation for further details on configuring Trust Chains.
2. Used as the service principal name of the delegating user when creating a Kerberos token. The service principal name can be determined by executing the Microsoft utility **setspn** (that is, **setspn -L *user***, where *user* is the identity of the user on the junctioned Web server).

Options

<service name>
The service name which is used to locate the trust chain within TFIM.

Usage

This stanza entry is required when TFIM SSO authentication is used over junctions.

Default value

None

```
service-name = HTTP/bigblue.wma.ibm.com
```

tfim-cluster-name

Syntax

```
tfim-cluster-name = name of cluster
```

Description

The name of the WebSphere cluster for the Federation Runtime service. The cluster is defined by this stanza entry along with a corresponding **[tfim-cluster:*<cluster>*]** stanza.

Options

name of cluster
The name of the WebSphere cluster that contains the Federation Runtime service.

Usage

This stanza entry is required when TFIM SSO authentication is used over junctions.

Default value

None

```
tfim-cluster-name = wascluster01
```

token-collection-size

Syntax

```
token-collection-size = number
```

Description

Specifies the number of security tokens for WebSEAL to retrieve from Federation Runtime in a single request. This construct is currently only supported for the Kerberos STS module.

Note: The number value for this stanza entry should be relatively low. Each token retrieved from Federation Runtime is quite large; specifying a large number dramatically increases the size of the packets received from TFIM, which in turn increases the size of the session and the amount of memory used by WebSEAL.

Options

number
The number of security tokens that WebSEAL retrieves from the Federation Runtime and cache for subsequent requests.

Usage

The `token-collection-size` stanza entry is required when the Federation SSO authentication is used over junctions.

Default value

None

Example

```
token-collection-size = 10
```

token-type

Syntax

```
token-type = token_type
```

Description

Specifies the type of token to be requested from the Federation Runtime. This value should correspond to the 'Token Type URI' field for the corresponding trust chain within TFIM.

Options

token_type
Indicates that the type of token to be requested from the Federation Runtime. Available options are Kerberos, SAML and LDAP.

Usage

This stanza entry is required when TFIM SSO authentication is used over junctions.

Default value

None

Example

```
token-type = http://docs.oasis-open.org/wss/oasis-wss-kerberos-token-profile-1.
1#GSS_Kerberosv5_AP_REQ
```

token-transmit-name

Syntax

```
token-transmit-name = text
```

Description

The name given to the security token within the junctioned Web server request.

Options

text
This is a free text field.

Usage

This stanza entry is required when TFIM SSO authentication is used over junctions.

Default value

None

Example

```
token-transmit-name = Authorization
```

token-transmit-type

Syntax

```
token-transmit-type = {header | cookie}
```

Description

The type of mechanism which will be used to transmit the security token to the junctioned Web server.

Options

header
The security token will be included in a header.

cookie
The security token will be included in a cookie.

Usage

This stanza entry is required when TFIM SSO authentication is used over junctions.

Default value

None

Example

```
token-transmit-type = header
```

[tfimsso:<jct-id>] stanza

always-send-tokens

Use the **always-send-tokens** stanza entry to control whether WebSEAL sends a security token for every HTTP request.

Syntax

```
always-send-tokens = {yes|true|no|false}
```

Description

Indicates whether WebSEAL sends a security token for every HTTP request or whether WebSEAL waits for a 401 response before it adds the security token.

You can use this configuration item to avoid generating and adding a security token to every request if the back-end web server can maintain user sessions. This configuration item is only useful if the request for authentication involves a 401 response, which currently applies to the Federation Runtime single sign-on only.

Options

yes
WebSEAL sends a security token for every HTTP request.

no
WebSEAL waits for a 401 response before it sends a security token for an HTTP request.

Usage

The `always-send-token` stanza entry is required when Federation SSO authentication is used over junctions.

Default value

None

Example

```
always-send-tokens = false
```

applies-to

Use the **applies-to** stanza entry to specify the location of the Security Token Service module if you are using Federation Runtime SSO authentication.

Syntax

```
applies-to = http://<webseal-server>/<junction>
```

Description

Path to specify the location to search for the appropriate Security Token Service (STS) module in Federation Runtime.

Options

http://<webseal-server>/<junction>
The host name or IP address of the WebSEAL server, along with the junction name. This address is similar to the URL that is used to access the junction.

Usage

The `applies-to` stanza entry is required when the Federation SSO authentication is used over junctions.

Default value

None

Example

```
applies-to = http://webseal-server/jct
```

one-time-token

Syntax

```
one-time-token = {true | false}
```

Description

This boolean value is used to indicate whether the security token that is produced by TFIM is only valid for a single transaction. An example of a one-time-token is a Kerberos token, which can only be used for a single authentication operation.

Usage

This stanza entry is required when TFIM SSO authentication is used over junctions.

Default value

True.

Example

```
one-time-token = false
```

preserve-xml-token

Syntax

```
preserve-xml-token = {true | false}
```

Description

This value controls whether to use the requested `BinarySecurityToken` XML structure in its entirety or whether only the encapsulated token should be used. Set this configuration entry to `true` only if the junctioned Web server understands and expects the `BinarySecurityToken` XML structure.

Usage

This stanza entry is required when TFIM SSO authentication is used over junctions.

Default value

True.

Example

```
preserve-xml-token = false
```

renewal-window

Syntax

```
renewal-window = number of seconds
```

Description

The length of time, in seconds, by which the expiration of security tokens will be reduced. This entry is used to make allowances for differences in system times and transmission times for the security tokens.

Options

number of seconds
Number of seconds by which the expiration of security tokens will be reduced to make allowances for differences between system times and transmission times for security tokens.

Usage

This stanza entry is required when TFIM SSO authentication is used over junctions.

Default value

None

Example

```
renewal-window = 15
```

service-name

Syntax

```
service-name = <servicename>
```

Description

1. Used by TFIM when searching for a matching trust chain. This configuration entry will be compared against the configured **AppliesTo** service name value for each trust chain. The second field within the **AppliesTo** service name configuration entry should be set to either asterisk (*) to match all service names, or it should be set to the value defined by this configuration item. See the TFIM documentation for further details on configuring Trust Chains.
2. Used as the service principal name of the delegating user when creating a Kerberos token. The service principal name can be determined by executing the Microsoft utility **setspn** (that is, **setspn -L *user***, where *user* is the identity of the user on the junctioned Web server).

Options

<service name>
The service name which is used to locate the trust chain within TFIM.

Usage

This stanza entry is required when TFIM SSO authentication is used over junctions.

Default value

None

```
service-name = HTTP/bigblue.wma.ibm.com
```

tfim-cluster-name

Syntax

```
tfim-cluster-name = name of cluster
```

Description

The name of the WebSphere cluster for the Federation Runtime service. The cluster is defined by this stanza entry along with a corresponding **[tfim-cluster:*<cluster>*]** stanza.

Options

name of cluster
The name of the WebSphere cluster that contains the Federation Runtime service.

Usage

This stanza entry is required when TFIM SSO authentication is used over junctions.

Default value

None

```
tfim-cluster-name = wascluster01
```

token-collection-size

Syntax

```
token-collection-size = number
```

Description

Specifies the number of security tokens for WebSEAL to retrieve from Federation Runtime in a single request. This construct is currently only supported for the Kerberos STS module.

Note: The number value for this stanza entry should be relatively low. Each token retrieved from Federation Runtime is quite large; specifying a large number dramatically increases the size of the packets received from TFIM, which in turn increases the size of the session and the amount of memory used by WebSEAL.

Options

number
The number of security tokens that WebSEAL retrieves from the Federation Runtime and cache for subsequent requests.

Usage

The `token-collection-size` stanza entry is required when the Federation SSO authentication is used over junctions.

Default value

None

Example

```
token-collection-size = 10
```

token-type

Syntax

```
token-type = token_type
```

Description

Specifies the type of token to be requested from the Federation Runtime. This value should correspond to the 'Token Type URI' field for the corresponding trust chain within TFIM.

Options

token_type
Indicates that the type of token to be requested from the Federation Runtime. Available options are Kerberos, SAML and LDAP.

Usage

This stanza entry is required when TFIM SSO authentication is used over junctions.

Default value

None

Example

```
token-type = http://docs.oasis-open.org/wss/oasis-wss-kerberos-token-profile-1.
1#GSS_Kerberosv5_AP_REQ
```

token-transmit-name

Syntax

```
token-transmit-name = text
```

Description

The name given to the security token within the junctioned Web server request.

Options

text
This is a free text field.

Usage

This stanza entry is required when TFIM SSO authentication is used over junctions.

Default value

None

Example

```
token-transmit-name = Authorization
```

token-transmit-type

Syntax

```
token-transmit-type = {header | cookie}
```

Description

The type of mechanism which will be used to transmit the security token to the junctioned Web server.

Options

header
The security token will be included in a header.

cookie
The security token will be included in a cookie.

Usage

This stanza entry is required when TFIM SSO authentication is used over junctions.

Default value

None

Example

```
token-transmit-type = header
```

[tfim-cluster:<*cluster*>] stanza

This stanza contains definitions for a particular cluster of Federation Runtime servers.

basic-auth-user

Use the **basic-auth-user** stanza entry to specify the user name to include in the basic authentication header when WebSEAL communicates with the Federation Runtime.

Syntax

```
basic-auth-user = <user_name>
```

Description

Specifies the name of the user for WebSEAL to include in the basic authentication header when it is communicating with the Federation Runtime server.

Options

<user_name>
The user name that WebSEAL includes in the basic authentication header.

Usage

This stanza entry is optional.

Note: Configure this entry if the Federation Runtime server is configured to require basic authentication.

Default value

None.

Example

```
basic-auth-user = user_name
```

basic-auth-passwd

Use the **basic-auth-passwd** stanza entry to specify the password that WebSEAL includes in the basic authentication header when it communicates with Tivoli Federated Identity Manager.

Syntax

```
basic-auth-passwd = <password>
```

Description

Specifies the password for WebSEAL to include in the basic authentication header when it is communicating with the Tivoli Federated Identity Manager server.

Options

<password>
The password that WebSEAL includes in the basic authentication header.

Usage

This stanza entry is optional.

Note: Configure this entry if the Tivoli Federated Identity Manager server is configured to require basic authentication.

Default value

None.

Example

```
basic-auth-passwd = password
```

gsk-attr-name

Syntax

```
gsk-attr-name = {enum | string | number}:id:value
```

Description

Specify additional GSKit attributes to use when initializing an SSL connection with the Federation Runtime. This configuration entry can be specified multiple times. Configure a separate entry for each GSKit attribute.

Options

{enum | string | number}
The GSKit attribute type.

id
The identity associated with the GSKit attribute.

value
The value for the GSKit attribute.

Usage

This stanza entry is optional.

See "Appendix: Supported GSKit attributes" on page 583 for a list of GSKit attributes that can be configured.

You cannot configure the following restricted GSKit attributes:

```
GSK_KEYRING_FILE
GSK_KEYRING_STASH_FILE
GSK_KEYRING_LABEL
GSK_CIPHER_V2
GSK_V3_CIPHER_SPECS
GSK_PROTOCOL_TLSV1
GSK_FIPS_MODE_PROCESSING
```

If you attempt to modify any of these attributes then an error message will be generated.

Default value

None.

Example

The following entry is for the GSKit attribute GSK_HTTP_PROXY_SERVER_NAME, which has an identity value of 225:

```
gsk-attr-name = string:225:proxy.ibm.com
```

See also

"gsk-attr-name" on page 78
"gsk-attr-name" on page 508
"jct-gsk-attr-name" on page 511

handle-idle-timeout

Syntax

```
handle-idle-timeout = <number>
```

Description

Specifies the length of time, in seconds, before an idle handle is removed from the handle pool cache.

Options

<number>
Length of time, in seconds, before an idle handle is removed from the handle pool cache.

Usage

This stanza entry is required when Kerberos authentication is used over junctions.

Default value

None

Example

```
handle-idle-timeout = 240
```

handle-pool-size

Syntax

```
handle-pool-size = <number>
```

Description

Specifies the maximum number of cached handles that WebSEAL uses when communicating with Federation Runtime.

Options

<number>
Maximum number of handles that WebSEAL caches to communicate with Federation Runtime.

Usage

This `handle-pool-size` stanza entry is required when the Federation SSO authentication is used over junctions.

Default value

`10`

Example

```
handle-pool-size = 10
```

load-balance

Controls the behavior when multiple servers with the same configured priority are available.

Syntax

```
load-balance = {yes | no}
```

Description

Specifies if round robin load balancing is used when multiple servers are configured with the same priority.

Options

yes
All requests are sent to matching servers in a round-robin fashion.

no
All requests are sent to the first matching server that is available. The matching order is the order they appear in the configuration file.

Usage

This stanza entry is optional.

Default value

The default value is `yes`.

Example

```
load-balance = yes
```

max-wait-time

Use this entry to control the maximum length of time, in seconds, that the request will block while waiting for a server to become available.

Syntax

```
max-wait-time = <number>
```

Description

Specifies the maximum length of time, in seconds, that the request will block while waiting for a server to become available. This configuration entry can be used to help eliminate errors being returned to the client during a server failover.

Options

<number>
Length of time, in seconds, that the request will block while waiting for a server to become available.

Usage

This stanza entry is optional.

Default value

0

Example

```
max-wait-time = 0
```

server

Syntax

```
server = {[0-9],}<URL>
```

Description

Specifies the priority level and URL for a single Federation Runtime server that is a member of the cluster identified for this **[tfim-cluster:<*cluster*>]** stanza.

Options

[*0-9*]
A digit, 0-9, that represents the priority of this server within the cluster (9 is the highest, 0 is the lowest). If the priority is not specified, a priority of 9 is assumed.

Note: There can be no space between the comma (,) and the URL. If no priority is specified, the comma is omitted.

<URL>
A well-formed HTTP or HTTPS uniform resource locator for the server.

Usage

The `server` stanza entry is required when the Federation SSO authentication is used over junctions.

Note: You can specify multiple server entries for a particular cluster for failover and load balancing.

Default value

None

Example

```
server = 9,http://tfim-server.example.com/TrustServerWST13/
 services/RequestSecurityToken
```

ssl-fips-enabled

Syntax

```
ssl-fips-enabled = {yes|no}
```

Description

Determines whether Federal Information Process Standards (FIPS) mode is enabled with the Federation Runtime.

Notes:

- If no configuration entry is present, the setting from the global setting, determined by the Verify Access policy server, takes effect.
- The **[tfim-cluster:<*cluster*>] ssl-nist-compliance** setting can override this entry. If **ssl-nist-compliance** is set to yes, FIPS mode processing is automatically enabled.

Options

yes
FIPS mode is enabled.

no
FIPS mode is disabled.

Usage

This stanza entry is required if both of the following conditions are true:

- One or more of the cluster **server** entries use SSL (that is, contains an HTTPS protocol specification in the URL).
- A certificate is required other than the default certificate used by WebSEAL when communicating with the policy server.

Note: If this entry is required, but it is not specified in the **[tfim-cluster:<*cluster*>]** stanza, WebSEAL uses the value in the global **[ssl]** stanza.

Default value

None.

Note: If you want to use a FIPS level that is different to the Verify Access policy server, edit the configuration file and specify a value for this entry.

Example

```
ssl-fips-enabled = yes
```

ssl-keyfile

Syntax

```
ssl-keyfile = <file_name>
```

Description

Specifies the name of the key database file that houses the client certificate for WebSEAL to use.

Options

<file_name>
Name of the key database file that contains the client-side certificate for WebSEAL to use when the Federation Runtime single sign-on is enabled for the junction.

Usage

This stanza entry is required if both of the following conditions are true:

- One or more of the cluster **server** entries use SSL (that is, contains an HTTPS protocol specification in the URL).
- A certificate is required other than the default certificate used by WebSEAL when communicating with the policy server.

Default value

None.

Example

```
ssl-keyfile = default-webseald.kdb
```

ssl-keyfile-label

Syntax

```
ssl-keyfile-label = <label-name>
```

Description

Specifies the label of the client-side certificate in the key database.

Options

<label-name>
Label of the client-side certificate in the key database.

Usage

This stanza entry is required if both of the following conditions are true:

- One or more of the cluster **server** entries use SSL (that is, contains an HTTPS protocol specification in the URL).
- A certificate is required other than the default certificate used by WebSEAL when communicating with the policy server.

Default value

None.

Example

```
ssl-keyfile-label = WebSEAL-Test
```

ssl-keyfile-stash

Syntax

```
ssl-keyfile-stash = <filename.sth>
```

Description

Specifies the name of the password stash file for the key database file.

Options

<filename.sth>
The name of the password stash file for the key database file.

Usage

This stanza entry is required if both of the following conditions are true:

- One or more of the cluster **server** entries use SSL (that is, contains an HTTPS protocol specification in the URL).
- A certificate is required other than the default certificate used by WebSEAL when communicating with the policy server.

Default value

None.

Example

```
ssl-keyfile-stash = default-webseald.sth
```

ssl-nist-compliance

Use the **ssl-nist-compliance** stanza entry in the **[tfim-cluster:<*cluster*>]** stanza to enable or disable NIST SP800-131A compliance when WebSEAL communicates with the Federation Runtime.

Syntax

```
ssl-nist-compliance = {yes|no}
```

Description

Enables or disables NIST SP800-131A compliance when WebSEAL communicates with Federation Runtime.

Enabling NIST SP800-131A compliance results in the following automatic configuration:

- Enables FIPS mode processing.

  Note: When NIST SP800-131A compliance is enabled, FIPS mode processing is enabled regardless of the setting for the **[tfim-cluster:<*cluster*>] ssl-fips-enabled** configuration entry.
- Enables TLS v1.2.

  Note: TLS v1 and TLS v1.1 are not disabled.
- Enables the appropriate signature algorithms.
- Sets the minimum RSA key size to 2048 bytes.

If this **ssl-nist-compliance** configuration entry is not present, WebSEAL uses the global **nist-compliance** setting in the **[ssl]** stanza.

Options

yes
A value of yes enables NIST SP800-131A compliance.

no
A value of no disables NIST SP800-131A compliance.

Usage

This stanza entry is optional.

Default value

no

Example

```
ssl-nist-compliance = no
```

ssl-valid-server-dn

Syntax

```
ssl-valid-server-dn = <DN-value>
```

Description

Specifies the distinguished name of the server, which is obtained from the server SSL certificate, that WebSEAL can accept.

Options

<DN-value>
The distinguished name of the server, which is obtained from the server SSL certificate, that WebSEAL accepts. If no value is specified, then WebSEAL considers all domain names valid. You can specify multiple domain names by including multiple **ssl-valid-server-dn** configuration entries.

Usage

This stanza entry is required if both of the following conditions are true:

- One or more of the cluster **server** entries use SSL (that is, contains an HTTPS protocol specification in the URL).
- A certificate is required other than the default certificate used by WebSEAL when communicating with the policy server.

Default value

None.

Example

```
ssl-valid-server-dn = CN=Verify Access,OU=SecureWay,O=Tivoli,C=US
```

timeout

Syntax

```
timeout = <number of seconds>
```

Description

Specifies the length of time, in seconds, to wait for a response from Federation Runtime.

Options

<number of seconds>
The length of time, in seconds, to wait for a response from Federation Runtime.

Usage

The `timeout` stanza entry is required when the Federation SSO authentication is used over junctions.

Default value

None.

Example

```
timeout = 240
```

[token] stanza

Use this stanza to define your token settings.

token-auth

Enables authentication by using the token authentication mechanism.

Syntax

```
token-auth = {none|http|https|both}
```

Description

This entry specifies which protocols are supported.

Options

{none|http|https|both}
The value `both` means both HTTP and HTTPS.

Usage

This stanza entry is required.

Default value

none

Example

```
token-auth = none
```

[user-agent] stanza

Use the **[user-agent]** stanza to specify a category name for a particular user-agent string in the HTTP Request header. WebSEAL uses the user-agent string to categorize the incoming requests to make flow data statistics more meaningful.

user-agent

Use the ***user-agent*** stanza entry to configure the category name that WebSEAL uses for a particular user-agent string in the HTTP Request header when it categorizes the incoming requests.

Syntax

```
user-agent = pattern
```

Description

When WebSEAL records flow data statistics, it can use the user-agent string in the HTTP Request header to categorize the incoming requests. Categorizing requests based on the user-agent can make the statistical data more useful.

Use this stanza to specify a list of category names and patterns for the user-agent strings to match. You can repeat a category so that multiple patterns match a single category. The patterns are evaluated in the order of their definition. WebSEAL selects the first match to categorize each request.

Note: The stanza must always end with an entry that contains the match-all pattern *.

Options

pattern

The appliance uses this pattern to categorize the incoming requests. The appliance categorizes each request by matching the user-agent string value in the HTTP Request header with the defined pattern list.

Note: The pattern can contain the wildcard characters * and ?. The patterns are not case-sensitive.

Usage

This stanza entry is optional.

Default value

None.

Example

In this example, both Android and iOS user-agent strings match the MOBILE category. WebSEAL uses the SUNDRY category if a user-agent string does not match any of the other defined patterns.

```
INTERNET_EXPLORER = *msie*
FIREFOX = *firefox*
CHROME = *chrome*
MOBILE = *android*
MOBILE = *ios*
SUNDRY = *
```

[user-agent-groups] stanza

Use the `[user-agent-groups]` stanza to map arbitrary user-agent strings to defined groups. These groups can then be referenced elsewhere in the configuration, namely in the '`[cookie-attributes]`' stanza.

group-name

Use the `group-name` stanza entry to configure the group name that WebSEAL uses for a particular user-agent string.

Syntax

```
group-name = pattern
```

Description

Use this stanza to specify a list of group names and patterns for the user-agent strings to match. You can repeat a group name so that multiple patterns match a single group. The patterns are evaluated in the order of their definition. These groups can then be referenced elsewhere in the configuration, namely in the '`[cookie-attributes]`' stanza.

Options

`pattern`
The user agent string which matches the user agents to be included in the group. The '*' and '?' pattern matching characters are supported.

Usage

This stanza entry is optional.

Default value

None

Example

In this example, a group is constructed which consists of user agents which are incompatible with the `SameSite=None` cookie attribute.

```
unsupported-same-site = *CPU iPhone OS 12*
unsupported-same-site = *iPad; CPU OS 12*
unsupported-same-site = *Macintosh; Intel Mac OS X 10_14*Safari*
unsupported-same-site = *Chrome/5*
unsupported-same-site = *Chrome/6*
```

[user-attribute-definitions] stanza

Use this stanza to modify the data type, the category, or both of a custom attribute.

Note: This stanza applies only to the IBM Security Verify Access for Mobile product.

attr_ID

Specify ***attr_ID*** entries to modify the data type, the category, or both of a custom attribute.

Syntax

```
attr_ID.datatype = data_type
attr_ID.category = category_name
```

Description

Use the appropriate stanza entry syntax depending on if you want to set the data type or category of a custom attribute from the default values.

Options

attr_ID
Specify the attribute identifier for which you want to set the data type or category. The attribute ID must match the name that exists in the `[azn-decision-info]` stanza entry.

datatype *data_type*
Set the data type of an attribute from the default of `string` to a specified *data_type*. Your choices are:

- `string`
- `boolean`
- `integer`
- `double`
- `time`
- `date`
- `x500name`

Table 6. Required date and time formats

Data type	Format	Example
Date	Date: *yyyy-mm-dd*	`2014-06-10`
	Date with timezone: *yyyy-mm-dd*`{+|-}`*hh:mm*	`2014-06-10+05:00`
Time	Time: *hh*`:`*mm*`:`*ss*	`13:00:01`
	Time with timezone: *hh*`:`*mm*`:`*ss*`{+|-}`*hh*`:`*mm*	`01:34:22-06:00`

category *category_name*
Set the category of an attribute from the default of `Environment` to a specified *category_name*. Your choices are:

- `Environment`
- `Subject`
- `Action`
- `Resource`

Usage

This stanza entry is not required.

The **[user-attribute-definitions]** stanza and the *attr_ID* entries apply only to Advanced Access Control.

Default value

The default value for data type is `string`.

The default value for category is `Environment`.

Example: Updating the data type for JSON data

If you defined a custom attribute in the `[azn-decision-info]` stanza as:

```
urn:example:company:txn:value = post-data:/"accountBalances"/"savings"
```

Then, you can set the data type of `urn:example:company:txn:value` to `double` by using the following stanza and entry:

```
[user-attribute-definitions]
urn:example:company:txn:value.datatype = double
```

Example: Updating the category for form data

If you defined a custom attribute in the `[azn-decision-info]` stanza as:

```
urn:example:company:txn:userid = post-data:userid
```

Then, you can set the category of `urn:example:company:txn:userid` to `Subject` by using the following stanza and entry:

```
[user-attribute-definitions]
urn:example:company:txn:userid.category = Subject
```

[user-map-authn] stanza

Use the **[user-map-authn]** stanza to define authenticated user mapping settings.

rules-file

Use the **rules-file** stanza entry to define the name of the rules file to be used by the authenticated user mapping module.

Syntax

```
rules-file = file-name
```

Description

The name of the rules file that is used by the authenticated user mapping module.

Options

file-name
Name of the rules file.

Usage

This stanza entry is required.

Default value

None.

Example

```
rules-file = auth-rules.txt
```

debug-level

Use the **debug-level** stanza entry to define the initial tracing level of the user mapping module.

Syntax

```
debug-level = level
```

Description

Controls the initial trace level for the user mapping module.

Options

level

The level variable that indicates the trace level of the user mapping module, with 1 designating a minimal amount of tracing and 9 designating the maximum amount of tracing.

Note: You can also use the Security Verify Access **pdadmin** trace commands to modify the trace level by using the trace component name of pd.cas.usermap. This trace component is only available after the first HTTP request is processed

Usage

This stanza entry is optional.

Default value

0

Note: A debug level of 0 results in no tracing output.

Example

```
debug-level = 0
```

[validate-headers] stanza

Use the **[validate-headers]** stanza to list those headers to be validated on each request.

hdr

Use the **hdr** stanza entry to define the header to be validated on each request.

Syntax

```
<hdr> = <value>
```

Description

Use this stanza entry to specify the header to be validated on each request. For example, to ensure that all requests are for www.ibm.com, set:

```
host = www.ibm.com
```

Multiple headers can be configured by specifying multiple stanza entries.

Options

<hdr>
Name of the header.

<value>
Value of the header.

Usage

This stanza entry is optional. If multiple headers of the same name are configured, the corresponding header in the request must match one of the configured values.

Default value

None.

Example

```
host = www.ibm.com
```

[websocket] stanza

Use the **[websocket]** stanza to define settings for WebsSocket support.

WebSocket is a standard that provides two-way asynchronous data transfer over a TCP or TLS connection.

max-worker-threads

Use the **max-worker-threads** entry to define the maximum number of threads that are used to proxy WebSocket connections through WebSEAL.

Syntax

```
max-worker-threads = number
```

Description

A value of zero blocks WebSockets from being proxied. Each WebSocket requires two worker threads. If the number of worker threads in use is more than the value of **max-worker-threads**, WebSEAL immediately closes the WebSocket even if the WebSocket upgrade request to the junction succeeds. These threads operate independently from the **[server] worker-threads**.

Options

number
The maximum number of threads that can be used to proxy WebSocket connections through WebSEAL.

Usage

This stanza entry is required.

Default value

```
0
```

Example

```
max-worker-threads = 0
```

idle-worker-threads

Use the **idle-worker-threads** entry to define the number of worker threads that can be left running idle.

Syntax

```
idle-worker-threads = number
```

Description

To avoid spending extra resource and time on starting and stopping WebSocket worker threads, the threads can be left running idle. Although keeping them alive and idle when not in use consumes memory resources, it saves CPU and thread start-up time when the threads are required.

Options

number
The number of threads that can be left running idle.

Usage

This stanza entry is required.

Default value

```
0
```

Example

```
idle-worker-threads = 0
```

jct-read-inactive-timeout

Use the **jct-read-inactive-timeout** entry to define the number of seconds to wait for data from the junctioned WebSocket server before the connection is closed.

Syntax

```
jct-read-inactive-timeout = seconds
```

Description

The **jct-read-inactive-timeout** entry defines the number of seconds to wait for data from the junctioned WebSocket server. If the timeout is reached, then the WebSocket connection is closed.

An inactive timeout occurs when the WebSocket application or browser attempts to read data from the socket, but none becomes available within the timeout period because the other end has not written any data to the socket. In other words, an inactive timeout is a read data operation timeout. This situation might occur because the writing end is hung or the underlying TCP/IP connection was ended without notification by a firewall. However, certain legitimate causes might also trigger the timeout. For example, during user interaction, the user takes a break before responding to the application, which might trigger this timeout.

Set this timeout value to be higher than any timeout that each WebSocket application might employ, allowing the WebSocket client and server components to deal with timeouts themselves. This option is useful in setting an upper limit to help ensure that malicious attacks or unexpected errors cannot permanently consume all the WebSEAL WebSockets resources.

Options

seconds
The number of seconds to wait for data from the junctioned WebSocket server before the WebSocket connection is closed.

Usage

This stanza entry is required.

Default value

```
120
```

Example

```
jct-read-inactive-timeout = 120
```

clt-read-inactive-timeout

Use the **clt-read-inactive-timeout** entry to define the number of seconds to wait for data from the WebSocket client before the connection is closed.

Syntax

```
clt-read-inactive-timeout = seconds
```

Description

The **clt-read-inactive-timeout** entry defines the number of seconds to wait for data from the WebSocket client or browser. If the timeout is reached, then the WebSocket connection is closed.

An inactive timeout occurs when the WebSocket application or browser attempts to read data from the socket, but none becomes available within the timeout period because the other end has not written any data to the socket. In other words, an inactive timeout is a read data operation timeout. This situation might occur because the writing end is hung or the underlying TCP/IP connection was ended without notification by a firewall. However, certain legitimate causes might also trigger the timeout. For example, during user interaction, the user takes a break before responding to the application, which might trigger this timeout.

Set this timeout value to be higher than any timeout that each WebSocket application might employ, allowing the WebSocket client and server components to deal with timeouts themselves. This option is useful in setting an upper limit to help ensure that malicious attacks or unexpected errors cannot permanently consume all the WebSEAL WebSockets resources.

Options

seconds
The number of seconds to wait for data from the WebSocket client before the connection is closed.

Usage

This stanza entry is required.

Default value

```
120
```

Example

```
clt-read-inactive-timeout = 120
```

jct-write-blocked-timeout

Use the **jct-write-blocked-timeout** entry to define the number of seconds to wait while WebSEAL is blocked sending data to the junctioned WebSocket server before the connection is closed.

Syntax

```
jct-write-blocked-timeout = seconds
```

Description

The **jct-write-blocked-timeout** entry defines the number of seconds to wait while WebSEAL is blocked sending data to the junctioned WebSocket server. If the timeout is reached, then the WebSocket connection is closed.

A blocked timeout occurs when the WebSocket application or browser writes to a socket, but the socket has a full write data queue due to the other end not reading the data. In other words, a blocked timeout is a write data operation timeout. This situation might occur because the reading end is hung. However, certain legitimate causes might also trigger the timeout. For example, the effective data bandwidth of the connection to the remote end is smaller than the rate at which data is being generated.

Set this timeout value to be higher than any timeout that each WebSocket application might employ, allowing the WebSocket client and server components to deal with timeouts themselves. This option is useful in setting an upper limit to help ensure that malicious attacks or unexpected errors cannot permanently consume all the WebSEAL WebSockets resources.

Options

seconds
The number of seconds to wait while WebSEAL is blocked sending data to the junctioned WebSocket server before the connection is closed.

Usage

This stanza entry is required.

Default value

```
120
```

Example

```
jct-write-blocked-timeout = 120
```

clt-write-blocked-timeout

Use the **clt-write-blocked-timeout** entry to define the number of seconds to wait while WebSEAL is blocked sending data to the WebSocket client before the connection is closed.

Syntax

```
clt-write-blocked-timeout = seconds
```

Description

The **clt-write-blocked-timeout** entry defines the number of seconds to wait while WebSEAL is blocked sending data to the WebSocket client or browser. If the timeout is reached, then the WebSocket connection is closed.

A blocked timeout occurs when the WebSocket application or browser writes to a socket, but the socket has a full write data queue due to the other end not reading the data. In other words, a blocked timeout is a write data operation timeout. This situation might occur because the reading end is hung. However, certain legitimate causes might also trigger the timeout. For example, the effective data bandwidth of the connection to the remote end is smaller than the rate at which data is being generated.

Set this timeout value to be higher than any timeout that each WebSocket application might employ, allowing the WebSocket client and server components to deal with timeouts themselves. This option is useful in setting an upper limit to help ensure that malicious attacks or unexpected errors cannot permanently consume all the WebSEAL WebSockets resources.

Options

seconds
The number of seconds to wait while WebSEAL is blocked sending data to the WebSocket client or browser before the connection is closed.

Usage

This stanza entry is required.

Default value

```
120
```

Example

```
clt-write-blocked-timeout = 120
```

[waf] stanza

Use the **[waf]** stanza to define which requests the Reverse Proxy will process with the web application firewall.

The reverse proxy incorporates the ModSecurity rules processing engine, which can be enabled on a per request basis.

request-match

Use the **request-match** entry to define which requests should be processed by the ModSecurity engine.

Syntax

```
request-match = [phases]<request-line>
```

Description

The request-match configuration entry is used to define the pattern to be matched against the HTTP request line, which includes method, URI, and protocol. If a match is successful, then the web application firewall rules engine processing is triggered.

Multiple entries can be specified if needed.

Options

phases

An optional list of ModSecurity phases for which the web application firewall rules engine processing will be triggered. Phases can be provided as a comma separated list or hyphenated range of numbers. The supported phases include:

Table 7. ModSecurity phases

Phase	
1	Request Headers
2	Request Body
3	Response Headers
4	Response Body
5	Logging

If a list of phases is not supplied, the WAF processing will be triggered for every phase.

request-line

Contains the request line to be matched against. The pattern matching is case-sensitive. Wildcard characters # and ? can be used.

Usage

This stanza entry is optional.

Default value

```
None
```

Example

```
request-match = GET /index.html HTTP/1.1
request-match = GET /jct/*
request-match = [1-2,5]GET /login/*
```

log-cfg

Syntax

```
log-cfg = agent [parameter=value],[parameter=value]...
```

Description

Configures the ModSecurity event logging. You can use the available parameters to configure the logging agents.

Options

agent

Specifies the logging agent. The agent controls the logging destination for server events. Valid agents include:

- `stdout`
- `stderr`
- `file`
- `remote`
- `rsyslog`

parameter

The different agents support the following configuration parameters:

Table 8. Logging agent configuration parameters

Parameter	**Supporting agents**
`buffer_size`	`remote`
`compress`	`remote`
`dn`	`remote`
`error_retry`	`remote, rsyslog`
`flush_interval`	`all`
`hi_water`	`all`
`log_id`	`file, rsyslog`
`max_event_len`	`rsyslog`
`mode`	`file`
`path`	`all`
`port`	`remote, rsyslog`
`queue_size`	`all`
`rebind_retry`	`remote, rsyslog`
`rollover_size`	`file`

Table 8. Logging agent configuration parameters (continued)	
Parameter	**Supporting agents**
`server`	`remote, rsyslog`
`ssl_keyfile`	`rsyslog`
`ssl_label`	`rsyslog`
`ssl_stashfile`	`rsyslog`

Note: For a complete description of the available logging agents and the supported configuration parameters, see the *IBM Security Verify Access for Web: Auditing Guide*.

Usage

This stanza entry is required.

Default value

None.

Example

To send web application firewall events to a file called `msg__waf.log`:

```
log-cfg = file path=msg__waf.log
```

To send web application firewall events to a remote syslog server:

```
log-cfg = rsyslog server=timelord,port=514,log_id=webseal-instance
```

Appendix: Supported GSKit attributes

You can configure the these GSKit attributes with Security Verify Access.

Strings

GSK_HTTP_PROXY_SERVER_NAME
Sets the http proxy server for http CDP CRL retrieval if required. The numeric identifier is 225.

GSK_SSL_EXTN_SERVERNAME_REQUEST
Sets the server name to be requested. The numeric identifier is 230.

GSK_SSL_EXTN_SERVERNAME_CRITICAL_REQUEST
Sets the server name to be requested. This request must be satisfied. If this request is not satisfied, an error is returned. The numeric identifier is 231.

GSK_HTTP_CRL_CACHE_SIZE
Sets the CRL cache size for the HTTP server. The value must be equal or more than zero. If the value is greater than zero enables HTTP CRL caching and sets the number of cached CRL's to be kept in cache to the value specified. The lifetime and validity of cache entries are maintained internally by using the `nextUpdate` and `max-age` caching directives. The cache size is maintained by replacing entries in a LRU manner.

Note: Keep the value small (32 or less) as CRLs can be large and therefore, consume large amounts of memory to remain in cache.

Enums

GSK_ALLOW_UNAUTHENTICATED_RESUME
The numeric identifier is 423. One of the following ENUM values must be specified (The default is GSK_ALLOW_UNAUTHENTICATED_RESUME_OFF):

GSK_ALLOW_UNAUTHENTICATED_RESUME_ON
Indicates that a session resume can be completed successfully even if the client has not provided a certificate during the initial handshake when the server is configured for client authentication. The numeric identifier is 588.

GSK_ALLOW_UNAUTHENTICATED_RESUME_OFF
Indicates that a session resume cannot be completed successfully when a client has not provided a certificate during the initial handshake when the server is configured for client authentication. This will cause the connection to complete an entire SSL handshake. This will ensure that server has the opportunity to authenticate the client. The numeric identifier is 589.

This ENUM_ID may only be set prior to gsk_environment_init().

GSK_SSL_SUITEB_MODE_PROCESSING
The numeric identifier is 454. One of the following ENUM values must be specified (The default is GSK_FALSE):

GSK_TRUE
SSL Suite B mode is set. The setting will restrict SSL session negotiation to only use TLS Suite B Profile; RFC 5430, approved mode of operation which restricts Cipher Suites, Certificates and Signature and Hash Algorithms. The numeric identifier is 1.

Note: This setting enables both 128 bit and 192 bit Security levels of Suite B. Do not make other settings related to CipherSuites, Protocol and Signature and Hash Algorithms once this setting has been made.

GSK_FALSE
SSL Suite B mode is not enabled. The numeric identifier is 0.

GSK_SSL_SUITEB_128BIT_MODE_PROCESSING
The numeric identifier is 455. One of the following ENUM values must be specified (The default is GSK_FALSE):

GSK_TRUE
SSL Suite B 128 bit Security mode is set. The setting will restrict SSL session negotiation to only use TLS Suite B Profile; RFC 5430, approved mode of operation which restricts Cipher Suites, Certificates and Signature and Hash Algorithms. The numeric identifier is 1.

Note: This setting enables only 128 bit Security level of Suite B. Do not make other settings related to CipherSuites, Protocol and Signature and Hash Algorithms once this setting has been made.

GSK_FALSE
SSL Suite B mode is not enabled. The numeric identifier is 0.

Note: This ENUM may only be set prior to gsk_environment_init(). FIPS-140 certified cryptographic modules should also be configured if using this setting. This setting will enable the TLS12 Protocol and disable all others.

GSK_SSL_SUITEB_192BIT_MODE_PROCESSING
The numeric identifier is 456. One of the following ENUM values must be specified (The default is GSK_FALSE):

GSK_TRUE
SSL Suite B 192 bit Security mode is set. The setting will restrict SSL session negotiation to only use TLS Suite B Profile; RFC 5430, approved mode of operation which restricts Cipher Suites, Certificates and Signature and Hash Algorithms. The numeric identifier is 1.

Note: This setting enables only 192 bit Security level of Suite B. Do not make other settings related to CipherSuites, Protocol and Signature and Hash Algorithms once this setting has been made.

GSK_FALSE
SSL Suite B mode is not enabled. The numeric identifier is 0.

GSK_LDAP_REQUIRED_AT_INIT
Specify the requirements of an LDAP server at environment initialization. The numeric identifier is 412. One of the following ENUM values must be specified (The default is GSK_INIT_CRL_LDAP_REQUIRED_OFF) :

GSK_INIT_CRL_LDAP_REQUIRED_ON
Operational LDAP server (CRL database) is required during environment initialization. The numeric identifier is 538.

GSK_INIT_CRL_LDAP_REQUIRED_OFF
Availability of an active LDAP server (CRL database) is not required during environment initialization. The numeric identifier is 539.

GSK_CC_MODE_CONTROL
This group controls the Common Criteria Mode operational requirements. The numeric identifier is 418. One of the following ENUM_VALUE values must be specified (The defaults is OFF for each of these):

GSK_CC_MODE_DISABLE_STASH_FILE_ON
Disable the use of stash files to open keystores. The numeric identifier is 555.

GSK_CC_MODE_DISABLE_STASH_FILE_OFF
Allow the use of stash files to open keystores. The numeric identifier is 556.

This ENUM may only be set prior to gsk_environment_init(). gsk_environment_init() will fail if the use of stash files have been disallowed but no keystore password has been given. It cannot be set using an environment variable.

GSK_CC_MODE_FIPS_ON
FIPS mode is set. The numeric value is 557. The enumerated value for GSK_BASE_CRYPTO_LIBRARY must not be GSK_BASE_CRYPTO_RSA (the default is GSK_BASE_CRYPTO_ICC) or an error is returned. This enum has the same effect as setting all of GSK_FIPS_MODE_PROCESSING_ON, GSK_SSL_FIPS_MODE_PROCESSING_ON, GSK_ICC_FIPS_MODE_PROCESSING_ON. Additionally setting this enum will have a similar effect to setting GSK_NIST_DES_FIPS_DEPRECATION except that the deprecation of DES will happen immediately and not wait until May 18 2007.

GSK_CC_MODE_FIPS_OFF
FIPS mode is not enabled. The numeric identifier is 558. This enum has the same effect as GSK_FIPS_MODE_PROCESSING_OFF. This ENUM may only be set prior to gsk_environment_init(). gsk_environment_init() will fail if FIPS mode is not supported on the platform. It cannot be set using an environment variable.

GSK_CC_MODE_ENFORCE_STRONG_PWD_ON
Enforce the use of Common Criteria strength passwords for keystore operations. The numeric identifier is 559.

GSK_CC_MODE_ENFORCE_STRONG_PWD_OFF
Remove the enforcement of the use of Common Criteria strength passwords for keystore operations. The numeric identifier is 560.

This ENUM may only be set prior to gsk_environment_init(). gsk_environment_init() will fail if the given password does not meet the strength rules. It cannot be set using an environment variable.

GSK_CC_MODE_DISABLE_PKCS11_ON
Disable the use of pkcs#11 devices. The numeric identifier is 561.

GSK_CC_MODE_DISABLE_PKCS11_OFF
Allow the use of pkcs#11 devices. The numeric identifier is 562.

This ENUM may only be set prior to gsk_environment_init(). It cannot be set using an environment variable.

GSK_CC_MODE_ENFORCE_STRONG_KDB_ON
Enforce that only newer version cms keystores that have stronger tamper protection be used. The numeric identifier is 563.

GSK_CC_MODE_ENFORCE_STRONG_KDB_OFF
Remove the enforcement that only newer version cms keystores that have stronger tamper protection be used. The numeric identifier is 564.

GSK_CC_MODE_STRICT_BASIC_CONST_ON
Enforce the rule that non end entity certificates that are missing the Basic Constraints extension are not permitted to be used in a validation chain. The numeric identifier is 565.

GSK_CC_MODE_STRICT_BASIC_CONST_OFF
Allow non end entity certificates that are missing the Basic Constraints extension to be permitted to be used in a validation chain. The numeric identifier is 566.

GSK_CC_MODE_ENFORCE_RIP_ON
Ensure that GSKit clears residual information for a session when that session encounters ssl errors. The numeric identifier is 567.

GSK_CC_MODE_ENFORCE_RIP_OFF
Do not enforce that GSKit clears residual information for a session when that session encounters ssl errors. The numeric identifier is 568.

GSK_NIST_DES_FIPS_DEPRECATION
On May 19 2007 NIST have determined that DES will no longer be a FIPS certified cipher. Turning this flag on will cause DES to be removed from the cipher list in FIPS mode after this date. The numeric identifier is 433.

GSK_TRUE
Turn DES deprecation on after May 18 2007. The numeric identifier is 1.

GSK_FALSE
Do not remove DES from the FIPS cipher list after May 18 2007. The numeric identifier is 0.

GSK_BINARY_DN_MATCHING_ENABLE
Allows for faster operation by comparing DN names using Binary DER Encoding The default is off (Disabled). The numeric identifier is 441.

GSK_TRUE
Turn Binary Matching On (Not recommended). The numeric identifier is 1.

GSK_FALSE
Turn Binary Matching Off. The numeric identifier is 0.

GSK_PROTOCOL_SSLV2
Enables or disables the SSL V2 protocol. Note that in FIPs mode of operation (see GSK_FIPS_MODE_PROCESSING) this setting will have no effect. The numeric identifier is 403. ENUM_VALUE must specify one of the following operations (The default is GSK_PROTOCOL_SSLV2_ON):

GSK_PROTOCOL_SSLV2_ON
Enable SSL V2

GSK_PROTOCOL_SSLV2_OFF
Disable SSL V2

GSK_PROTOCOL_SSLV3
Enables or disables the SSL V3 protocol. The numeric identifier is 404. ENUM_VALUE must specify one of the following operations (The default is GSK_PROTOCOL_SSLV3_ON):

GSK_PROTOCOL_SSLV3_ON
Enable SSL V3

GSK_PROTOCOL_SSLV3_OFF
Disable SSL V3

GSK_PROTOCOL_TLSV10
Enables or disables the TLSV10 protocol. The numeric identifier is 436. ENUM_VALUE must specify one of the following operations (The default is on):

GSK_TRUE
Enable TLSV10

GSK_FALSE
Disable TLSV10

GSK_PROTOCOL_TLSV11
Enables or disables the TLSV11 protocol. The numeric identifier is 437. ENUM_VALUE must specify one of the following operations (The default is on):

GSK_TRUE
Enable TLSV11

GSK_FALSE
Disable TLSV11

GSK_PROTOCOL_TLSV12
Enables or disables the TLSV12 protocol. The numeric identifier is 438. ENUM_VALUE must specify one of the following operations (The default is on):

GSK_TRUE
Enable TLSV12

GSK_FALSE
Disable TLSV12

GSK_V2_CIPHER_SPECS
If multiple connections occur under a SSL session the values set for this field may not be used. The cipher specification negotiated during the first SSL connection of a session will be used until that session expires. Here is the list of available cipher specs. The list contains the string values that can be used with the buf_value for this buffer ID. Any combination of these may be used; none may be used twice.

- 1-RC4 US
- 2-RC4 Export
- 3-RC2 US
- 4-RC2 Export
- 6-DES 56-Bit
- 7-Triple DES US

If a NULL string ("") is specified for the cipherspec list, SSL version 2 protocols will not be used.

The default cipherspec is "713642". The numeric identifier is 205.

GSK_V3_CIPHER_SPECS_EX, GSK_TLSV10_CIPHER_SPECS_EX, GSK_TLSV11_CIPHER_SPECS_EX, GSK_TLSV12_CIPHER_SPECS_EX
Allows the user to specify Cipher Specs for TLS protocol versions. The numeric identifiers are 240, 241, 242, and 243. Different TLS Protocols may have mutually exclusive Cipher Spec.

The buffer cotains a list of comma delimted string values that are defined by RFC 2246, 4346, 5246, 4492, 5289.

Example : Setting AES TLS Ciphersuite would require a buffer comtaining
« TLS_RSA_WITH_AES_128_CBC_SHA, TLS_RSA_WITH_AES_256_CBC_SHA”

Numbers

GSK_LDAP_FAILOVER_RECONNECTION_PERIOD
If multiple LDAP servers are specified for the failover function and the first LDAP server on the list is not available, the next one on the list will be queried until one is available or all the LDAP servers

are tried. Periodically, an attempt will be made to retry the LDAP server query process. This attribute specifies the time period before the query retry is to begin. The value of int_value must be in the range of 0 - 86400 seconds. Defaults:

- For a single LDAP server, 0 seconds.
- When multiple LDAP servers are specified, 300 seconds.

The numeric identifier is 307.

GSK_V2_SIDCACHE_SIZE
The number of entries in the SID (Session ID) cache used for SSLV2, range 0-2047 (default=256). The numeric identifier is 304.

GSK_V3_SIDCACHE_SIZE
The number of entries in the SID (Session ID) cache used for SSLV3 and TLSV1, range 0-MAXINT (default=512). This setting does not impose an upper limit, however GSKit internally imposes a limit that may be reviewed over time. Currently the internal limit is 655360. Note: Very large cache sizes could have adverse impacts on process performance due to the large memory usage. The cache memory allocation is dynamic in that memory is not allocated for cache entries until they are required, thus the memory usage may in fact be far less than the maximum number of cache entries specified. It is suggested that application consider these aspects when setting the cache size. The numeric identifier is 305.

GSK_OCSP_TIMEOUT
Sets the timeout in seconds that we will wait for a response from the server. The default is 30. The numeric identifier is 318.

GSK_HTTP_CDP_MAX_RESPONSE_SIZE
Sets the maximum size in bytes that GSKit will accept as a response from a HTTP Server when retrieving a CRL. This may help protect against a denial of service attack. The default is 204800 (200K). The numeric identifier is 316.

GSK_HTTP_CDP_TIMEOUT
Sets the timeout in seconds that we will wait for a response from the server. The default is 30. The numeric identifier is 319.

GSK_MAX_SSL_MESSAGE_SIZE
Sets the maximum message size that can be received by GSKit. This setting is design to protect against certain Denial of Service attack where very lare message can be used to exhust memory on a system. The default is 128K bytes. The numeric identifier is 320.

GSK_HTTP_PROXY_SERVER_PORT
Sets the http proxy server port for http CDP CRL retrieval if needed. The numeric identifier is 317.

GSK_LDAP_SERVER_VERSION
Sets the LDAP protocol version to be used. This should be set to 2 or 3. The numeric identifier is 314.

GSK_V2_SESSION_TIMEOUT
SSL V2 session time-out. int_value must be in the range 0-100 seconds (default=100). The numeric identifier is 301.

GSK_V3_SESSION_TIMEOUT
SSL V3 session time-out. int_value must be in the range 0-86400 seconds (default=86400, 24 hours). The numeric identifier is 302.

GSK_ALLOW_ONLY_EXTENDED_RENEGOTIATION
The numeric identifier is 447. The default value is GSK_TRUE. This setting can only be used in the [ssl] section.

GSK_TRUE
Only RFC5746 Renegotiation Allowed. The numeric identifier is 1.

GSK_FALSE
All TLS Renegotiation disabled . The numeric identifier is 0.

Index

D

E

F

G

H

I

J

K

L

M

N

O

P

R

S

T

U

V

W

X