Web Authentication Enhancements—Customizing Authentication Proxy Web Pages

The Web Authentication Enhancements—Customizing Authentication Proxy Web Pages feature allows you to display four HTML pages to users in place of the switch's internal default HTML pages during web-based authentication. The four pages are Login, Success, Fail, and Expire.

This module also describes the following enhancement features for the custom HTML pages:

- Custom Web Authentication Result Display Enhancement feature—ensures that the authentication results display on the main HTML page.
- Support for Custom Web Authentication Download Bundle feature—ensures that one or more custom HTML pages can be downloaded and configured from a single tar file bundle.
- Virtual IP Support for Images in Custom Web Authentication feature—ensures that users can configure a virtual IP address.

- Finding Feature Information, page 1
- Prerequisites for Customizing Authentication Proxy Web Pages, page 2
- Restrictions for Customizing Authentication Proxy Web Pages, page 2
- Information About Customizing Authentication Proxy Web Pages, page 2
- How to Configure Custom Authentication Proxy Web Pages, page 4
- Configuration Examples for Customization of Authentication Proxy Web Pages, page 11
- Additional References, page 13
- Feature Information for Customization of Authentication Proxy Web Pages, page 14

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To

find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Customizing Authentication Proxy Web Pages

- To enable the custom web pages feature, you must specify all four custom HTML files.
  - If fewer than four files are specified, the internal default HTML pages are used.
  - The four custom HTML files must be present on the disk or flash of the switch.
- Any external link from a custom page requires the configuration of an intercept ACL within the admission rule.
- Any name resolution required for external links or images requires the configuration of an intercept ACL within the admission rule to access a valid DNS server.

Restrictions for Customizing Authentication Proxy Web Pages

- If the custom web pages feature is enabled, a configured auth-proxy-banner cannot be used.
- If the custom web pages feature is enabled, the redirect URL for successful login feature is not available.

Information About Customizing Authentication Proxy Web Pages

Custom Authentication Proxy Web Pages

The switch's internal HTTP server hosts four custom HTML pages (in addition to the four default internal HTML pages) for delivery to an authenticating client during the web-based authentication process. These four pages allow the server to notify the user of the following four states of the authentication process:

- Login—User credentials are requested.
- Success—Login is successful.
- Fail—Login has failed.
- Expire—The login session has expired due to excessive login failures.

You can substitute your custom HTML pages for the four default internal HTML pages or you can specify a URL to which the user is redirected after a successful authentication; effectively replacing the internal success page.

This module describes two methods by which you can configure custom web pages:

- Using the **ip admission proxy** command.

- Using the **parameter-map type webauth** command for identity control policy-based access session management.

Images for Custom Web Pages

This section describes the guidelines for all images on the custom web pages:

- An image file has a size limit of 256 KB.
- All image files must have a filename that begins with "web_auth_" (such as "web_auth_logo.jpg" instead of "logo.jpg").

Note The Virtual IP Support for Images in Custom Web Authentication feature supports image filenames that do not require any prefix. Users can specify any image name.

- All image filenames must be less than 63 characters.
- Any images on the custom pages must be located on an accessible HTTP server. An intercept ACL must be configured within the admission rule to allow access to the HTTP server.

Result Display Enhancement

The Custom Web Authentication Result Display Enhancement feature displays the authentication results on the main HTML page. There is no pop-up window to display the authentication results.

Custom Web Authentication Download Bundle

The Support for Custom Web Authentication Download Bundle feature ensures that one or more custom HTML pages can be downloaded and configured from a single tar file bundle. The images and the custom pages containing the images are also part of the same downloadable tar file bundle.

Virtual IP Support for Images

The Virtual IP Support for Images in Custom Web Authentication feature supports the following enhancements:

- Image filenames do not require any prefix. Users can specify any image name.
- Users need not specify the wireless management interface IP address to indicate the source of the image in the HTML code. Instead, users can configure the **virtual-ip** command in parameter-map webauth configuration mode (config-params-parameter-map) and specify a virtual IP address. For more information about the **virtual-ip** command, see the "Configuring a Parameter Map for Custom Authentication Proxy Web Pages" section or the *Cisco IOS Security Command Reference: Commands S to Z*.

Parameter Map for Custom Authentication Proxy Web Pages

A parameter map allows you to modify parameters that control the behavior of actions configured under a control policy. A parameter map for web-based authentication sets parameters that can be applied to subscriber sessions during authentication. If you do not create a parameter map, the policy uses default parameters.

How to Configure Custom Authentication Proxy Web Pages

Configuring Custom Authentication Proxy Web Pages

To specify the use of your custom authentication proxy web pages, first store your custom HTML files on the switch's internal disk or flash memory and then perform this task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip admission proxy http login page file** *device*:*login-filename*
4. **ip admission proxy http success page file** *device*:*success-filename*
5. **ip admission proxy http failure page file** *device*:*fail-filename*
6. **ip admission proxy http expired page file** *device*:*expired-filename*
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip admission proxy http login page file** *device*:*login-filename* **Example:** `Device(config)# ip admission proxy http login page file disk1:login.htm`	Specifies the location in the switch memory file system of the custom HTML file to be used in place of the default login page. • The device: is either disk or flash memory, such as disk0:.

	Command or Action	Purpose
Step 4	**ip admission proxy http success page file** *device*:*success-filename* **Example:** `Device(config)# ip admission proxy http success page file disk1:success.htm`	Specifies the location of the custom HTML file to be used in place of the default login success page.
Step 5	**ip admission proxy http failure page file** *device*:*fail-filename* **Example:** `Device(config)# ip admission proxy http failure page file disk1:fail.htm`	Specifies the location of the custom HTML file to be used in place of the default login failure page.
Step 6	**ip admission proxy http expired page file** *device*:*expired-filename* **Example:** `Device(config)# ip admission proxy http expired page file disk1:expired.htm`	Specifies the location of the custom HTML file to be used in place of the default login expired page.
Step 7	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Specifying a Redirect URL for Successful Login

Before You Begin

Note You can specify a URL to which the user will be redirected upon successful authentication, effectively replacing the internal Success HTML page.

- If the custom authentication proxy web pages feature is enabled, the redirection URL feature is disabled and will not be available. You can perform redirection in the custom login success page.
- If the redirection URL feature is enabled, a configured auth-proxy-banner will not be used.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip admission proxy http success redirect** *url-string*
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip admission proxy http success redirect** *url-string* **Example:** `Device(config)# ip admission proxy http success redirect www.company.com`	Specifies a URL for redirection of the user in place of the default login success page.
Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring a Parameter Map for Custom Authentication Proxy Web Pages

Perform the following steps to define either a global or named parameter map for web-based authentication.

Note The configuration commands available in the global parameter map differ from the commands available in a named parameter map.

Before You Begin

Ensure that you configure a parameter map for identity control policy-based access session management.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **parameter-map type webauth** {*parameter-map-name* | **global**}
4. **banner** {**file** *location*:*filename* | **text** *banner-text*}
5. **consent email**
6. **custom-page** {**failure** | **login [expired]** | **success**} **device** *location*:*filename*
7. **max-http-conns** *number*
8. **redirect** {{**for-login** | **on-failure** | **on-success**} *url* | **portal** {**ipv4** *ipv4-address* | **ipv6** *ipv6-address*}}
9. **timeout init-state sec** *seconds*
10. **type** {**authbypass** | **consent** | **webauth** | **webconsent**}
11. **timeout fin-wait msec** *milliseconds*
12. **virtual-ip** {**ipv4** *ipv4-address* | **ipv6** *ipv6-address*}
13. **watch-list** {**add-item** {**ipv4** *ipv4-address* | **ipv6** *ipv6-address*} | **dynamic-expiry-timeout** *minutes* | **enabled**}
14. **end**
15. **show ip admission status** [**banners** | **custom-pages** | **parameter-map** [*parameter-map*]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**parameter-map type webauth** {*parameter-map-name*	**global**} **Example:** `Device(config)# parameter-map type webauth MAP-2`	Creates a parameter map and enters parameter-map webauth configuration mode. • The specific configuration commands supported for a global parameter map defined with the **global** keyword differ from the commands supported for a named parameter map defined with the *parameter-map-name* argument.
Step 4	**banner** {**file** *location*:*filename*	**text** *banner-text*} **Example:** `Device(config-params-parameter-map)# banner file flash:webauth_banner.html`	(Optional) Displays a banner on the web-authentication login web page.

	Command or Action	Purpose
Step 5	**consent email** **Example:** `Device(config-params-parameter-map)# consent email`	(Optional) Requests a user's e-mail address on the web-authentication login web page. • This command is supported in named parameter maps only.

Step 6	**custom-page** {**failure** | **login [expired]** | **success**} **device** *location*:*filename*
	
	Example:
	`Device(config-params-parameter-map)# custom-page login device flash:webauth_login.html`
	`Device(config-params-parameter-map)# custom-page login expired device flash:webauth_expire.html`
	`Device(config-params-parameter-map)# custom-page success device flash:webauth_success.html`
	`Device(config-params-parameter-map)# custom-page failure device flash:webauth_fail.html`
		(Optional) Displays custom authentication proxy web pages during web-based authentication.
		• You must configure all four custom HTML files. If fewer than four files are configured, the internal default HTML pages will be used.

Step 7	**max-http-conns** *number* **Example:** `Device(config-params-parameter-map)# max-http-conns 5`	(Optional) Limits the number of HTTP connections for each web authentication client.

Step 8	**redirect** {{**for-login** | **on-failure** | **on-success**} *url* | **portal** {**ipv4** *ipv4-address* | **ipv6** *ipv6-address*}}
	
	Example:
	`Device(config-params-parameter-map)# redirect portal ipv6 FE80::1`
	`Device(config-params-parameter-map)# redirect on-failure http://10.10.3.34/~sample/failure.html`
		(Optional) Redirects users to a particular URL during web-based authentication.

Step 9	**timeout init-state sec** *seconds* **Example:** `Device(config-params-parameter-map)# timeout init-state sec 60`	(Optional) Sets the initial state timeout, in seconds, for web-based authentication sessions.
Step 10	**type** {**authbypass** | **consent** | **webauth** | **webconsent**} **Example:** `Device(config-params-parameter-map)# type consent`	(Optional) Defines the methods supported by a web-based authentication parameter map. • This command is supported only for named parameter maps.
Step 11	**timeout fin-wait msec** *milliseconds* **Example:** `Device(config-params-parameter-map)# timeout fin-wait msec 30`	(Optional) Sets the TCP finish (FIN) packet timeout for web-based authentication sessions.

	Command or Action	Purpose
Step 12	**virtual-ip** {**ipv4** *ipv4-address* | **ipv6** *ipv6-address*} **Example:** `Device(config-params-parameter-map)# virtual-ip ipv6 FE80::1`	(Optional) Specifies a virtual IP address for web-based authentication clients. • This command is supported only for global parameter maps.

Step 13	**watch-list** {**add-item** {**ipv4** *ipv4-address* | **ipv6** *ipv6-address*} | **dynamic-expiry-timeout** *minutes* | **enabled**}
	
	Example:
	`Device(config-params-parameter-map)# watch-list enabled`
	`Device(config-params-parameter-map)# watch-list dynamic-expiry-timeout 20`
	`Device(config-params-parameter-map)# watch-list add-item ipv6 FE80::1`
		(Optional) Enables a watch list of web-based authentication clients.
		• This command is supported only for global parameter maps.

Step 14	**end** **Example:** `Device(config-params-parameter-map)# end`	(Optional) Exits parameter-map webauth configuration mode and returns to privileged EXEC mode.
Step 15	**show ip admission status** [**banners** | **custom-pages** | **parameter-map** [*parameter-map*]] **Example:** `Device# show ip admission status custom-pages`	(Optional) Displays information about configured banners and custom pages.

Verifying the Configuration of Parameter Maps for Custom Authentication Proxy Web Pages

Verifying the Configuration of a Global Parameter Map

Perform this task to verify the configuration of a global parameter map for custom authentication proxy web pages.

SUMMARY STEPS

1. **enable**
2. **show parameter-map type webauth global**

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode.

Example:

```
Device> enable
```

Step 2 **show parameter-map type webauth global**
Displays the configuration of a global parameter map for custom authentication proxy web pages.

Example:

```
Device# show parameter-map type webauth global

Parameter Map Name                                : global
  Type                                            : none
  Custom Page:
    Auth-proxy login                              : flash:login.html
  Auth-proxy Init State time                      : 120 sec
  Auth-proxy Fin Wait time                        : 3000 milliseconds
  Webauth max-http connection                     : 30
  Webauth logout-window                           : Enabled
  Consent Email                                   : Disabled
  Virtual-ipv4                                    : 3.3.3.10
  Webauth intercept https                         : Disabled
  Watch-list:
    Enabled                                       : no
  Webauth login-auth-bypass:
```

Verifying the Configuration of a Named Parameter Map

Perform this task to verify the configuration of a named parameter map for custom authentication proxy web pages.

SUMMARY STEPS

1. **enable**
2. **show parameter-map type webauth** *parameter-map-name*

DETAILED STEPS

Step 1 **enable**
Enables privileged EXEC mode.

Example:

```
Device> enable
```

Step 2 **show parameter-map type webauth** *parameter-map-name*
Displays the configuration of a named parameter map (webauth-name1) for custom authentication proxy web pages.

Example:

```
Device# show parameter-map type webauth webauth-name1

Parameter Map Name                              : webauth-name1
  Type                                          : webauth
  Auth-proxy Init State time                    : 120 sec
  Auth-proxy Fin Wait time                      : 3000 milliseconds
  Webauth max-http connection                   : 30
  Webauth logout-window                         : Enabled
  Consent Email                                 : Disabled
  Webauth login-auth-bypass:
```

Configuration Examples for Customization of Authentication Proxy Web Pages

Example: Configuring Custom Authentication Web Pages

```
Device> enable
Device# configure terminal
Device(config)# ip admission proxy http login page file disk1:login.htm
Device(config)# ip admission proxy http success page file disk1:success.htm
Device(config)# ip admission proxy http failure page file disk1:fail.htm
Device(config)# ip admission proxy http expired page file disk1:expired.htm
Device(config)# end
```

Example: Configuring a Redirection URL for Successful Login

```
Device> enable
Device# configure terminal
Device(config)# ip admission proxy http success redirect www.company.com
Device(config)# end
```

Example: Configuring Parameter Maps for Custom Authentication Proxy Web Pages

Global Parameter Map

```
Device> enable
Device# configure terminal
Device(config)# parameter-map type webauth global
Device(config-params-parameter-map)# timeout init-state sec 60
Device(config-params-parameter-map)# logging enabled
Device(config-params-parameter-map)# watch-list enabled
Device(config-params-parameter-map)# virtual-ip ipv6 FE80::1
Device(config-params-parameter-map)# redirect on-failure
http://10.10.3.34/~sample/failure.html
Device(config-params-parameter-map)# max-http-conns 100
Device(config-params-parameter-map)# watch-list dynamic-expiry-timeout 5000
Device(config-params-parameter-map)# banner file flash:webauth_banner.html
Device(config-params-parameter-map)# end
```

Named Parameter Map for Web Authentication Using Custom Pages

The following example shows how to configure a named parameter map for web authentication that defines custom pages for the login process, along with a control policy that uses the parameter map.

```
Device> enable
Device# configure terminal
Device(config)# parameter-map type webauth CUSTOM-WEBAUTH-MAP
Device(config-params-parameter-map)# type webauth
Device(config-params-parameter-map)# custom-page login device flash:login_page.html
Device(config-params-parameter-map)# custom-page success device flash:success_page.html
Device(config-params-parameter-map)# custom-page failure device flash:fail_page.html
Device(config-params-parameter-map)# custom-page login expired device flash:expire_page.html
Device(config-params-parameter-map)# exit
Device(config)# policy-map type control subscriber CUSTOM-WEBAUTH-POLICY
Device(config-event-control-policymap)# event session-started match-all
Device(config-class-control-policymap)# 10 class always do-until-failure
Device(config-action-control-policymap)# 10 authenticate using webauth parameter-map
CUSTOM-WEBAUTH-MAP retries 2 retry-time 0
Device(config-action-control-policymap)# end
```

Named Parameter Map for Consent Using Custom Pages

The following example shows how to configure a named parameter map for custom consent, along with the corresponding control policy that uses the parameter map:

```
Device> enable
Device# configure terminal
Device(config)# parameter-map type webauth CUSTOM-CONSENT-MAP
Device(config-params-parameter-map)# type consent
Device(config-params-parameter-map)# custom-page login device flash:consent_login_page.html
Device(config-params-parameter-map)# custom-page success device
flash:consent_success_page.html
Device(config-params-parameter-map)# custom-page failure device flash:consent_fail_page.html
Device(config-params-parameter-map)# custom-page login expired device
flash:consent_expire_page.html
Device(config-params-parameter-map)# end
Device(config)# ip access-list extended GUEST-ACL
Device(config-ext-nacl)# permit ip any 172.30.30.0 0.0.0.255
Device(config-ext-nacl)# permit ip any host 172.20.249.252
```

```
Device(config-ext-nacl)# exit
Device(config)# service-template GUEST-POLICY
Device(config-service-template)# access-group GUEST-ACL
Device(config-service-template)# exit
Device(config)# policy-map type control subscriber CUSTOM-CONSENT-POLICY
Device(config-event-control-policymap)# event session-started match-all
Device(config-class-control-policymap)# 10 class always do-until-failure
Device(config-action-control-policymap)# 10 authenticate using webauth parameter-map
CUSTOM-CONSENT-MAP
Device(config-action-control-policymap)# exit
Device(config-event-control-policymap)# event authentication-success match-all
Device(config-class-control-policymap)# 10 class always do-until-failure
Device(config-action-control-policymap)# 10 activate service-template GUEST-POLICY
Device(config-action-control-policymap)# end
```

Named Parameter Map for Web Authentication with Consent Using Custom Pages

The following example shows how to configure a named parameter map for web authentication with custom consent, along with the corresponding control policy that uses the parameter map:

```
Device> enable
Device# configure terminal
Device(config)# parameter-map type webauth CUSTOM-WEBAUTH-CONSENT-MAP
Device(config-params-parameter-map)# type webconsent
Device(config-params-parameter-map)# custom-page login device
flash:webauth_consent_login_page.html
Device(config-params-parameter-map)# custom-page success device
flash:webauth_consent_success_page.html
Device(config-params-parameter-map)# custom-page failure device
flash:webauth_consent_fail_page.html
Device(config-params-parameter-map)# custom-page login expired device
flash:webauth_consent_expire_page.html
Device(config-params-parameter-map)# exit
Device(config)# ip access-list extended GUEST-ACL
Device(config-ext-nacl)# permit ip any 172.30.30.0 0.0.0.255
Device(config-ext-nacl)# permit ip any host 172.20.249.252
Device(config-ext-nacl)# exit
Device(config)# service-template GUEST-POLICY
Device(config-service-template)# access-group GUEST-ACL
Device(config-service-template)# exit
Device(config)# policy-map type control subscriber CUSTOM-WEBAUTH-CONSENT-POLICY
Device(config-event-control-policymap)# event session-started match-all
Device(config-class-control-policymap)# 10 class always do-until-failure
Device(config-action-control-policymap)# 10 authenticate using webauth parameter-map
CUSTOM-WEBAUTH-CONSENT-MAP
Device(config-action-control-policymap)# exit
Device(config-event-control-policymap)# event authentication-success match-all
Device(config-class-control-policymap)# 10 class always do-until-failure
Device(config-action-control-policymap)# 10 activate service-template GUEST-POLICY
Device(config-action-control-policymap)# end
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases

Related Topic	Document Title
Security commands	• Cisco IOS Security Command Reference: Commands A to C • Cisco IOS Security Command Reference: Commands D to L • Cisco IOS Security Command Reference: Commands M to R • Cisco IOS Security Command Reference: Commands S to Z
Authentication, authorization, and accounting	*Authentication, Authorization, and Accounting (AAA) Configuration Guide*
Access lists and the Cisco IOS Firewall	"Access Control Lists: Overview and Guidelines" module of the *Security Configuration Guide: Access Control Lists* publication
Configuring identity control policies	*Identity-Based Networking Services Configuration Guide*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Customization of Authentication Proxy Web Pages

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 1: Feature Information for Customization of Authentication Proxy Web Pages

Feature Name	Releases	Feature Information
Web Authentication Enhancements—Customization of Authentication Proxy Web Pages	Cisco IOS 15.0(1)EX Cisco IOS XE 3.2SE	The Customization of Authentication Proxy Web Pages feature allows you to provide four HTML pages to be displayed to the user in place of the switch's internal default HTML pages during web-based authentication. The four pages are Login, Success, Fail, and Expire. In Cisco IOS XE Release 3.2SE, this feature is supported on the following platforms: • Catalyst 3850 Series Switches • Cisco 5760 Wireless LAN Controller
Custom Web Authentication Result Display Enhancement	Cisco IOS XE 3.6E	The Custom Web Authentication Result Display Enhancement feature displays the authentication results on the main HTML page. There is no pop-up window to display the authentication results. In Cisco IOS XE Release 3.6E, this feature is supported on the following platforms: • Catalyst 3850 Series Switches • Catalyst 3650 Series Switches

Feature Name	Releases	Feature Information
Support for Custom Web Authentication Download Bundle	Cisco IOS XE 3.6E	The Support for Custom Web Authentication Download Bundle feature ensures that one or more custom HTML pages can be downloaded and configured from a single tar file bundle. The images and the custom pages containing the images are also part of the same downloadable tar file bundle. In Cisco IOS XE Release 3.6E, this feature is supported on the following platforms: • Catalyst 3850 Series Switches • Catalyst 3650 Series Switches
Virtual IP Support for Images in Custom Web Authentication	Cisco IOS XE 3.6E	The Virtual IP Support for Images in Custom Web Authentication feature supports image filenames without prefixes and removes the requirement of users having to specify the wireless management interface IP address to indicate the source of the image in the HTML code. In Cisco IOS XE Release 3.6E, this feature is supported on the following platforms: • Catalyst 3850 Series Switches • Catalyst 3650 Series Switches