

Authentication and Authorization

This topic contains the following sections:

- Overview of Acquire End-User Credentials, on page 1
- Authentication Best Practices, on page 2
- Authentication Planning, on page 3
- Authentication Realms, on page 13
- Authentication Sequences, on page 33
- Failed Authentication, on page 35
- Credentials, on page 41
- Troubleshooting Authentication, on page 43

Overview of Acquire End-User Credentials

Server Type/Realm	Authentication Scheme	Supported Network Protocol	Notes
Active Directory	Kerberos NTLMSSP Basic	HTTP, HTTPS Native FTP, FTP over HTTP SOCKS (Basic authentication)	Kerberos is only supported in Standard mode. It is not supported in Cloud Connector mode.
LDAP	Basic	HTTP, HTTPS Native FTP, FTP over HTTP SOCKS	—

Authentication Task Overview

Step	Task	Links to Related Topics and Procedures
1	Create an authentication realm.	• How to Create an Active Directory Authentication Realm (NTLMSSP and Basic), on page 18 • Creating an LDAP Authentication Realm, on page 21

Step	Task	Links to Related Topics and Procedures
2	Configure global authentication settings.	• Configuring Global Authentication Settings, on page 26
3	Configure external authentication. You can authenticate users through an external LDAP or RADIUS server.	• External Authentication, on page 14
4	(Optional) Create and order additional authentication realms. Create at least one authentication realm for each authentication protocol and scheme combination you plan to use.	• Creating Authentication Sequences, on page 34
5	(Optional) Configure credential encryption.	• Configuring Credential Encryption, on page 43
6	Create Identification Profiles to classify users and client software based on authentication requirements.	• Classifying Users and Client Software
7	Create policies to manage Web requests from the users and user groups for which you created Identification Profiles.	• Managing Web Requests Through Policies Best Practices

Authentication Best Practices

- Create as few Active Directory realms as is practical. Multiple Active Directory realms require additional memory usage for authentication.
- If using NTLMSSP, authenticate users using either the Secure Web Appliance or the upstream proxy server, but not both. (Recommend Secure Web Appliance)
- If using Kerberos, authenticate using the Secure Web Appliance.
- For optimal performance, authenticate clients on the same subnet using a single realm.
- Some user agents are known to have issues with machine credentials or authentication failures, which can negatively impact normal operations. You should bypass authentication with these user agents. See Bypassing Authentication with Problematic User Agents , on page 36.
- Actively authenticating a client is a resource-intensive task. Authentication surrogates can be used to improve authentication performance by remembering an authenticated user for a set duration (default is 3600 seconds and configurable under (**Global Authentication** > **Surrogate Timeout**) after authentication has completed. IP surrogates should be used whenever possible to limit the number of active authentication events.
- For authentication setups with trusted Active Directory (AD), the authentication fails if the DC resolutions for the trusted domain is not successful. This is because of the introduction Samba 4.11.15, and it requires the trusted domain lookups to be successful. If the DC name resolution continues to fail, you must navigate

to **Network** > **Authentication** > **Authentication Server and Type and Scheme(s)** and uncheck the **Enable Trusted Domain Health Check** check box.

Authentication Planning

- Active Directory/Kerberos, on page 3
- Active Directory/Basic, on page 4
- Active Directory/NTLMSSP, on page 5
- LDAP/Basic, on page 5
- Identifying Users Transparently, on page 6

Active Directory/Kerberos

Explicit Forward	Transparent, IP-Based Caching	Transparent, Cookie-Based Caching
Advantages: • Better performance and interoperability when compared to NTLM • Works with both Windows and non-Windows clients that have joined the domain • Supported by all browsers and most other applications • RFC-based • Minimal overhead (Reauthentication is not required) • Works for HTTPS (CONNECT) requests • Because the passphrase is not transmitted to the authentication server, it is more secure • Connection is authenticated, not the host or IP address • Achieves true single sign-on in an Active Directory environment when the client applications are configured to trust the Secure Web Appliance	**Advantages:** • Better performance and interoperability when compared to NTLM • Works with both Windows and non-Windows clients that have joined the domain • Works with all major browsers • With user agents that do not support authentication, users only need to authenticate first in a supported browser • Relatively low overhead • Works for HTTPS requests if the user has previously authenticated with an HTTP request	**Advantages**: • Better performance and interoperability when compared to NTLM • Works with both Windows and non-Windows clients that have joined the domain • Works with all major browsers • Authentication is associated with the user rather than the host or IP address **Disadvantages**: • Each new web domain requires the entire authentication process because cookies are domain specific • Requires cookies to be enabled • Does not work for HTTPS requests

Active Directory/Basic

Explicit Forward	Transparent, IP-Based Caching	Transparent, Cookie-Based Caching
Advantages: • Supported by all browsers and most other applications • RFC-based • Minimal overhead • Works for HTTPS (CONNECT) requests • Because the passphrase is not transmitted to the authentication server, it is more secure • Connection is authenticated, not the host or IP address • Achieves true single sign-on in an Active Directory environment when the client applications are configured to trust the Secure Web Appliance **Disadvantages**: • Passphrase sent as clear text (Base64) for every request • No single sign-on • Moderate overhead: each new connection needs to be re-authenticated • Primarily supported on Windows only and with major browsers only	**Advantages:** • Works with all major browsers • With user agents that do not support authentication, users only need to authenticate first in a supported browser • Relatively low overhead • Works for HTTPS requests if the user has previously authenticated with an HTTP request **Disadvantages**: • Authentication credentials are associated with the IP address, not the user (does not work in Citrix and RDP environments, or if the user changes IP address) • No single sign-on • Passphrase is sent as clear text (Base64)	**Advantages**: • Works with all major browsers • Authentication is associated with the user rather than the host or IP address **Disadvantages**: • Each new web domain requires the entire authentication process because cookies are domain specific • Requires cookies to be enabled • Does not work for HTTPS requests • No single sign-on • Passphrase is sent as clear text (Base64)

Active Directory/NTLMSSP

Explicit Forward	Transparent
Advantages: • Because the passphrase is not transmitted to the authentication server, it is more secure • Connection is authenticated, not the host or IP address • Achieves true single sign-on in an Active Directory environment when the client applications are configured to trust the Secure Web Appliance **Disadvantages:** • Moderate overhead: each new connection needs to be re-authenticated • Primarily supported on Windows only and with major browsers only	**Advantages:** • More Flexible Transparent NTLMSSP authentication is similar to transparent Basic authentication except that the Web Proxy communicates with clients using challenge and response instead of basic clear text username and passphrase. The advantages and disadvantages of using transparent NTLM authentication are the same as those of using transparent Basic authentication except that transparent NTLM authentication has the added advantaged of not sending the passphrase to the authentication server and you can achieve single sign-on when the client applications are configured to trust the Secure Web Appliance.

LDAP/Basic

Explicit Forward	Transparent
Advantages: • RFC-based • More browser support than NTLM • Minimal overhead • Works for HTTPS (CONNECT) requests **Disadvantages:** • No single sign-on • Passphrase sent as clear text (Base64) for every request **Workarounds:** • Failed Authentication, on page 35	**Advantages:** • More Flexible than explicit forward. • More browser support than NTLM • With user agents that do not support authentication, users only need to authenticate first in a supported browser • Relatively low overhead • Works for HTTPS requests if the user has previously authenticated with an HTTP request **Disadvantages:** • No single sign-on • Passphrase is sent as clear text (Base64) • Authentication credentials are associated with the IP address, not the user (does not work in Citrix and RDP environments, or if the user changes IP address) **Workarounds:** • Failed Authentication, on page 35

Identifying Users Transparently

Traditionally, users are identified and authenticated by prompting them to enter a user name and passphrase. These credentials are validated against an authentication server, and then the Web Proxy applies the appropriate policies to the transaction based on the authenticated user name.

However, you can configure the Secure Web Appliance to authenticate users transparently—that is, without prompting the end user for credentials. Transparent identification authenticates the user by means of credentials obtained from another trusted source, with the assumption that the user has already been authenticated by that trusted source, and then applies the appropriate policies.

You might want to identify users transparently to:

- Create a single sign-on environment so users are not aware of the presence of a proxy on the network.
- To apply authentication-based policies to transactions coming from client applications that are incapable of displaying an authentication prompt to end users.

Identifying users transparently only affects how the Web Proxy obtains the user name and assigns an Identification Profile. After it obtains the user name and assigns an Identification Profile, it applies all other policies normally, regardless of how it assigned the Identification Profile.

If transparent authentication fails, you can configure how to handle the transaction: you can grant the user guest access, or you can force an authentication prompt to appear to the user.

When an end user is shown an authentication prompt due to failed transparent user identification, and the user then fails authentication due to invalid credentials, you can choose whether to allow the user guest access.



Note When you enable re-authentication and a transaction is blocked by URL filtering, an end-user notification page appears with the option to log in as a different user. Users who click the link are prompted for authentication. For more information, see Failed Authorization: Allowing Re-Authentication with Different Credentials, on page 39.

Understanding Transparent User Identification

The available methods of transparent user identification are:

- **Transparently identify users with ISE** – Available when the Identity Services Engine (ISE) or Passive Identity Connector (ISE-PIC) service is enabled (Network > Identity Services Engine). For these transactions, the user name and associated Secure Group Tags will be obtained from an Identity Services Engine server. If you are using ISE-PIC, the user name and associated ISE Secure Groups will be obtained. See Tasks for Integrating the ISE/ISE-PIC Service.
- **Transparently identify users with ASA** – Users are identified by the current IP address-to-user name mapping received from a Cisco Adaptive Security Appliance (for remote users only). This option is available when AnyConnect Secure Mobility is enabled and integrated with an ASA. The user name will be obtained from the ASA, and associated directory groups will be obtained from the authentication realm or sequence specified on the Secure Web Appliance. See Remote Users.
- **Transparently identify users with authentication realms** – This option is available when one or more authentication realms are configured to support transparent identification using one of the following authentication servers:

- **Active Directory** – Create an NTLM or Kerberos authentication realm and enable transparent user identification. In addition, you must deploy a separate Active Directory agent such as Cisco's Context Directory Agent. For more information, see Transparent User Identification with Active Directory, on page 7.
- **LDAP** – Create an LDAP authentication realm configured as an eDirectory, and enable transparent user identification. For more information, see Transparent User Identification with LDAP, on page 8.

AsyncOS for Web communicates at regular intervals with eDirectory or an Active Directory agent to maintain mappings that match authenticated user names to their current IP addresses.

Transparent User Identification with Active Directory

Active Directory does not record user log-in information in a format that is easily queried by other systems such as the Secure Web Appliance. Active Directory agents, such as Cisco's Context Directory Agent (CDA), are necessary to query the Active Directory security event logs for information about authenticated users.



Note CDA is not supported by Active Directory in Windows Server 2016. You can use Identity Services Engine (ISE) or ISE Passive Identity Controller (ISE-PIC) service to receive user information and achieve transparent user identification. You must configure identification profiles and relevant access policies, and decryption policies that use CDA, with ISE/ISE-PIC information when you switch from CDA to ISE/ISE-PIC.

AsyncOS for Web communicates with the Active Directory agent to maintain a local copy of the IP-address-to-user-name mappings. When AsyncOS for Web needs to associate an IP address with a user name, it first checks its local copy of the mappings. If no match is found, it queries an Active Directory agent to find a match.

For more information on installing and configuring an Active Directory agent, see the section "Setting Up an Active Directory Agent to Provide Information to the Secure Web Appliance" below.

Consider the following when you identify users transparently using Active Directory:

- Transparent user identification with Active Directory works with an NTLM or Kerberos authentication scheme only. You cannot use it with an LDAP authentication realm that corresponds to an Active Directory instance.
- Transparent user identification works with the versions of Active Directory supported by an Active Directory agent.
- You can install a second instance of an Active Directory agent on a different machine to achieve high availability. When you do this, each Active Directory agent maintains IP-address-to-user-name mappings independently of the other agent. AsyncOS for Web uses the backup Active Directory agent after three unsuccessful ping attempts to the primary agent.
- The Active Directory agent uses on-demand mode when it communicates with the Secure Web Appliance.
- The Active Directory agent pushes user log-out information to the Secure Web Appliance. Occasionally, some user log-out information is not recorded in the Active Directory security logs. This can happen if the client machine crashes, or if the user shuts down the machine without logging out. If there is no user log-out information in the security logs, an Active Directory agent cannot inform the appliance that the IP address no longer is assigned to that user. To obviate this possibility, you can define how long AsyncOS caches the IP-address-to-user mappings when there are no updates from an Active Directory agent. For

more information, see Using the CLI to Configure Advanced Transparent User Identification Settings, on page 10.

- The Active Directory agent records the `sAMAccountName` for each user logging in from a particular IP address to ensure the user name is unique.
- The client IP addresses that the client machines present to the Active Directory server and the Secure Web Appliance must be the same.
- AsyncOS for Web searches only direct parent groups for a user. It does not search nested groups.

Setting Up an Active Directory Agent to Provide Information to the Secure Web Appliance

Because AsyncOS for Web cannot obtain client IP addresses directly from Active Directory, it must obtain IP-address-to-user-name mapping information from an Active Directory agent.

Install an Active Directory agent on a machine in the network that is accessible to the Secure Web Appliance, and which can communicate with all visible Windows domain controllers. For best performance, this agent should be physically as close as possible to the Secure Web Appliance. In smaller network environments, you may want to install the Active Directory agent directly on the Active Directory server.



Note The Active Directory agent instance used to communicate with the Secure Web Appliance can also support other appliances, including Cisco's Adaptive Security Appliance and other Secure Web Appliances.

Obtaining, Installing, and Configuring Cisco's Context Directory Agent

You can find information about downloading, installing, and configuring the Cisco Context Directory Agent here: http://www.cisco.com/en/US/docs/security/ibf/cda_10/Install_Config_guide/cda10.html.



Note The Secure Web Appliance and Active Directory agent communicate with each other using the RADIUS protocol. The appliance and the agent must be configured with the same shared secret to obfuscate user passphrases. Other user attributes are not obfuscated.

Transparent User Identification with LDAP

AsyncOS for Web can communicate with an eDirectory server configured as a Lightweight Directory Access Protocol (LDAP) realms maintaining IP-address-to-user-name mappings. When a user logs in through an eDirectory client, the user is authenticated against the eDirectory server. When authentication succeeds, the client IP address is recorded in the eDirectory server as an attribute (NetworkAddress) of the user who logged in.

Consider the following when you identify users transparently using LDAP (eDirectory):

- The eDirectory client must be installed on each client workstation, and end users must use it to authenticate against an eDirectory server.
- The LDAP tree used by the eDirectory client log-in must be the same LDAP tree configured in the authentication realm.
- If the eDirectory clients use multiple LDAP trees, create an authentication realm for each tree, and then create an authentication sequence that uses each LDAP authentication realm.

- When you configure the LDAP authentication realm as an eDirectory, you must specify a Bind DN for the query credentials.
- The eDirectory server must be configured to update the NetworkAddress attribute of the user object when a user logs in.
- AsyncOS for Web searches only direct parent groups for a user. It does not search nested groups.
- You can use the NetworkAddress attribute for an eDirectory user to determine the most-recent log-in IP address for the user.

Rules and Guidelines for Transparent User Identification

Consider the following rules and guidelines when using transparent user identification with any authentication server:

- When using DHCP to assign IP addresses to client machines, ensure the IP-address-to-user-name mappings are updated on the Secure Web Appliance more frequently than the DHCP lease. Use the `tuiconfig` CLI command to update the mapping update interval. For more information, see Using the CLI to Configure Advanced Transparent User Identification Settings, on page 10.
- If a user logs out of a machine and another user logs into the same machine before the IP-address-to user-name mapping is updated on the Secure Web Appliance, then the Web Proxy logs the client as the previous user.
- You can configure how the Web Proxy handles transactions when transparent user identification fails. It can grant users guest access, or it can force an authentication prompt to appear to end users.
- When a user is shown an authentication prompt due to failed transparent user identification, and the user then fails authentication due to invalid credentials, you can choose whether to allow the user guest access.
- When the assigned Identification Profile uses an authentication sequence with multiple realms in which the user exists, AsyncOS for Web fetches the user groups from the realms in the order in which they appear in the sequence.
- When you configure an Identification Profile to transparently identify users, the authentication surrogate must be IP address. You cannot select a different surrogate type.
- When you view detailed transactions for users, the Web Tracking page shows which users were identified transparently.
- You can log which users were identified transparently in the access and WC3 logs using the `%m` and `x-auth-mechanism` custom fields. A log entry of `SSO_TUI` indicates that the user name was obtained by matching the client IP address to an authenticated user name using transparent user identification. (Similarly, a value of `SSO_ASA` indicates that the user is a remote user and the user name was obtained from a Cisco ASA using AnyConnect Secure Mobility.)

Configuring Transparent User Identification

Configuring transparent user identification and authorization is detailed in Overview of Acquire End-User Credentials, on page 1. The basic steps are:

- Create and order authentication realms.
- Create Identification Profiles to classify users and client software.
- Create policies to manage web requests from the identified users and user groups.

Using the CLI to Configure Advanced Transparent User Identification Settings

AsyncOS for Web provides the following TUI-related CLI commands:

- **`tuiconfig`** – Configure advanced settings associated with transparent user identification. Batch mode can be used to configure multiple parameters simultaneously.
  - **`Configure mapping timeout for Active Directory agent`** – Length of time, in minutes, IP-address-to-user mappings are cached for IP addresses retrieved by the AD agent when there are no updates from the agent.
  - **`Configure proxy cache timeout for Active Directory agent`** – Length of time, in seconds, proxy-specific IP-address-to-user mappings are cached; valid values range from five to 1200 seconds. The default and recommended value is 120 seconds. Specifying a lower value may negatively affect proxy performance.
  - **`Configure mapping timeout for Novell eDirectory`** – Length of time, in seconds, IP-address to-user mappings are cached for IP addresses retrieved from the eDirectory server when there are no updates from the server.
  - **`Configure query wait time for Active Directory agent`** – The length of time, in seconds, to wait for a reply from the Active Directory agent. When the query takes more than this value, transparent user identification is considered to have failed. This limits the authentication delay experienced by the end user.
  - **`Configure query wait time for Novell eDirectory`** – The length of time, in seconds, to wait for a reply from the eDirectory server. When the query takes more than this value, transparent user identification is considered to have failed. This limits the authentication delay experienced by the end user.

The Active Directory settings apply to all AD realms using an AD agent for transparent user identification. The eDirectory settings apply to all LDAP realms using eDirectory for transparent user identification.

If validation fails for any one parameter, none of the values will be changed.

- **`tuistatus`** – This command provides the following AD-related subcommands:
  - **`adagentstatus`** – Displays the current status of all AD agents, as well as information about their connections with the Windows domain controllers.
  - **`listlocalmappings`** – Lists all IP-address-to-user-name mappings stored on the Secure Web Appliance, as retrieved by the AD agent(s). It does not list entries stored on the agent(s), nor does it list mappings for which queries are currently in progress.

Configuring Single-Sign-on

Obtaining credentials transparently facilitates a single-sign-on environment. Transparent user identification is an authentication realm setting.

For Internet Explorer, be sure the Redirect Hostname is the short host name (containing no dots) or the NetBIOS name rather than a fully qualified domain. Alternatively, you can add the appliance host name to Internet Explorer's Local intranet zone (Tools > Internet options > Security tab); however, this will be required on every client. For more information about this, see How do I properly set up NTLM with SSO (credentials sent transparently)?

With Firefox and other non-Microsoft browsers, the parameters **network.negotiate-auth.delegation-uris**, **network.negotiate-auth.trusted-uris** and **network.automatic-ntlm-auth.trusted-uris** must be set to the transparent-mode Redirect Hostname. You also can refer to Firefox is not sending authentication credentials transparently (SSO). This article provides general information about changing Firefox parameters.

For information about the Redirect Hostname, see Configuring Global Authentication Settings, on page 26, or the CLI command `sethostname`.

Creating a Service Account in Windows Active Directory for Kerberos Authentication in High Availability Deployments

Use this procedure if you are having issues with high availability with Kerberos authentication. Scenarios, where issues may arise when using Kerberos Authentication in High Availability Deployments are:

- The servicePrincipalName of the high availability hostname is added to multiple machine accounts in the Active Directory.
- Kerberos authentication works if the servicePrincipalName has been added to single machine account in the Active Directory. When the primary node changes, high availability may be impacted, because different appliance nodes use different encryption strings to decrypt Kerberos service tickets.

Before you begin

- Choose the user name to be used for high availability with Kerberos authentication. We recommend creating a new user name, which will be used solely for this purpose.
- If you prefer using an existing user name:
  - Set a password, if the user name does not have one.
  - In the user account properties dialog box (in Active Directory users and computers):

    Ensure that the User must change password at next logon check box is unchecked.

    Check the Password never expires check box.

Procedure

Step 1 Create a new user name in Active Directory users and computers.

- Specify a password.
- Uncheck the User must change password at next logon check box.
- Check the Password never expires check box.

Step 2 Check if the SPN of the high availability hostname is associated with the Active Directory user object created or chosen. SPN consists of a http/ prefix, and is suffixed with the appliance's high availability hostname. Ensure that the clients are able to resolve the hostname.

a. Use the `setspn -q` command in Windows, to query for any existing association.

Example: `setspn -q http/highavail.com`

In this example, highavail.com is the appliance's high availability hostname.

b. Remove, or add the SPN depending on the results of the query:

Note
Kerberos HA service account passwords can only include letters, numbers, spaces and characters ~ ! @ # % ^ & () _ - {} ' / [] : ; , | + = * ? <>. If any of these 3 special characters $, `, or " are used in the Kerberos HA service account password, it will result in a failure during pre-authentication from both GUI and CLI. However, authentication is successful with all kinds of characters used in the password.

Query Result	Action
No such SPN found.	Associate the SPN of the high availability hostname is associated with the Active Directory user object. • Use the `setspn -s` command: `setspn -s http/highavail.com hausername` In this example, highavail.com is the appliance's high availability hostname, and hausername is the user name created or chosen.
Existing SPN found! The common name (CN) shows the user name created or chosen. Example: CN = hausername	No further action is necessary in the Active Directory.

Existing SPN found!
The common name (CN) does not show the user name created or chosen.
	a. Remove the SPN.
	Use the `setspn -d` command:
	`setspn -d http/highavail.com johndoe`
	In this example, highavail.com is the appliance's high availability hostname, and johndoe is the user name to be disassociated.
	b. Add the SPN.
	Use the `setspn -s` command:
	`setspn -s http/highavail.com hausername`
	In this example, highavail.com is the appliance's high availability hostname, and hausername is the user name created or chosen.

Note
Ensure that keytab authentication is enabled in the relevant Active Directory realm. See Creating an Active Directory Realm for Kerberos Authentication Scheme, on page 14. For realms already created, edit the realm, and enable the keytab authentication.

Authentication Realms

Authentication realms define the details required to contact the authentication servers and specify which authentication scheme to use when communicating with clients. AsyncOS supports multiple authentication realms. Realms can also be grouped into authentication sequences that allow users with different authentication requirements to be managed through the same policies.

Authentication Failover

In the current realm setup, there are one primary AD or LDAP and two backup servers. If the first primary server is not reachable, the query reaches to the first backup server. If the first backup server is also not reachable, the query reaches to the second server.

Table 1: Failover time using IPFW rule

Failover time	Failover time taken from primary to secondary backup in seconds
To break the connection between primary AD and Secure Web Appliance	75 to 80
To break the connection between primary AD and Secure Web Appliance and also to break connection between first backup and Secure Web Appliance	180 to 250
Reboot primary AD	42 secs
Power off Primary AD	75 to 80
Power off Primary AD and first backup server	180 to 250

If more than one servers are down, Secure Web Appliance retries to establish connection until a working domain controller is found.

- External Authentication, on page 14
- Creating an Active Directory Realm for Kerberos Authentication Scheme, on page 14
- How to Create an Active Directory Authentication Realm (NTLMSSP and Basic), on page 18
- Creating an LDAP Authentication Realm, on page 21
- About Deleting Authentication Realms, on page 26
- Configuring Global Authentication Settings, on page 26

Related Topics

- Authentication Sequences, on page 33
- RADIUS User Authentication

External Authentication

You can authenticate users through an external LDAP or RADIUS server.

Configuring External Authentication through an LDAP Server

Before you begin

Create an LDAP authentication realm and configure it with one or more external authentication queries. Creating an LDAP Authentication Realm, on page 21.

Procedure

Step 1 Enable external authentication on the appliance:

a) Navigate to **System Administration** > **Users**.

b) Click **Enable** in the External Authentication section.

c) Configure the options:

Option	Description
Enable External Authentication	—
Authentication Type	Select LDAP.
External Authentication Cache Timeout	The number of seconds AsyncOS stores the external authentication credentials before contacting the LDAP server again to re-authenticate. Default is zero (0).
LDAP External Authentication Query	A query configured with the LDAP realm.
Timeout to wait for valid response from server.	The number of seconds AsyncOS waits for a response to the query from the server.
Group Mapping	For each group name in the directory, assign a role.

Step 2 Submit and commit your changes.

Enabling RADIUS External Authentication

See Enabling External Authentication Using RADIUS.

Creating an Active Directory Realm for Kerberos Authentication Scheme

Before you begin

- Ensure that the appliance is configured in Standard mode (not Cloud Connector Mode).

- If you are setting up high availability, ensure that you also enable the **Use keytab authentication** check box in the Kerberos High Availability section, specified in **step 9**.

  If your appliance resides behind a HTTP/HTTPS traffic distribution device like a load balancer, you must associate the SPN of the traffic distribution device in the Active Directory with a user account, and enter the credentials of that user account in the Kerberos High Availability section. The SPN of the first device that redirects traffic in the network topology should be added. For example, if client devices' outbound network traffic passes through a traffic manager, a load balancer, and then to the Secure Web Appliance, the SPN for the traffic manager should be added to a user account on the Active Directory, and the user credentials should be entered in this section. This is because the traffic manager is the first device that encounters client devices' traffic.

- Prepare the Active Directory Server.
  - Install Active Directory on one of these servers: Windows server 2003, 2008, 2008R2, 2012, 2016 (for coeus 11.8, 12.0, 12.5, 14.0, and 14.5), or 2019 (for coeus 14.5 only).

    You can install Active Directory Windows server 2019 for coeus 12.5.
  - Create a user on the Active Directory server:
    - Create a user on the Active Directory server that is a member of the Domain Admins or Account Operators group.

      Or
    - Create a user name with the following permissions:
      - Active Directory permissions Reset Password
      - Validated write to servicePrincipalName
      - Write account restrictions
      - Write dNShost name
      - Write servicePrincipalName

        These are the minimal Active Directory permissions required by a user name to join an appliance to the domain and ensure its complete functioning.
  - Join your client to the domain. Supported clients are Windows XP, Windows 10 and Mac OS 10.5+.
  - Use the kerbtray tool from the Windows Resource Kit to verify the Kerberos ticket on the client: http://www.microsoft.com/en-us/download/details.aspx?id=17657.
  - Ticket viewer application on Mac clients is available under main menu > KeyChain Access to view the Kerberos tickets.
- Ensure that you have the rights and domain information needed to join the Secure Web Appliance to the Active Directory domain you want to authenticate against.
- Compare the current time on the Secure Web Appliance with the current time on the Active Directory server and verify that the difference is no greater than the time specified in the "Maximum tolerance for computer clock synchronization" option on the Active Directory server.

- If the Secure Web Appliance is managed by a Security Management appliance, be prepared to ensure that same-named authentication realms on different Secure Web Appliances have identical properties defined on each appliance.
- Secure Web Appliance configuration:
  - In explicit mode, the Secure Web Appliance host name (CLI command `sethostname`) and the proxy name configured in the browser must be the same.
  - In transparent mode, the Secure Web Appliance host name must be the same as the Redirect Hostname (see Configuring Global Authentication Settings, on page 26). Further, the Secure Web Appliance host name and Redirect Hostname must be configured prior to creating a Kerberos realm.
- Be aware that after you commit the new realm, you cannot change a protocol of realm authentication.
- Note that Single Sign On (SSO) must be configured on client browsers; see Configuring Single-Sign-on, on page 10.
- To simplify use of logs, customize the access log to use the %m custom field parameter. See Customizing Access Logs.

Note Kerberos HA service account passwords can only include letters, numbers, spaces and characters ~ ! @ # % ^ & () _ - {} ' / [] : ; , | + = * ? <>. If any of these 3 special characters $, `, or " are used in the Kerberos HA service account password, it will result in a failure during pre-authentication from both GUI and CLI. However, authentication is successful with all kinds of characters used in the password.

Procedure

Step 1 In the Cisco Secure Web Appliance web interface, choose **Network > Authentication**.

Step 2 Click **Add Realm**.

Step 3 Assign a unique name to the authentication realm using only alphanumeric and space characters.

Step 4 Select **Active Directory** in the Authentication Protocol field.

Step 5 Enter up to three fully-qualified domain names or IP addresses for the Active Directory server(s).

Example: `ntlm.example.com` .

An IP address is required only if the DNS servers configured on the appliance cannot resolve the Active Directory server hostname.

When multiple authentication servers are configured in the realm, the appliance attempts to authorize with up to three authentication servers before failing to authorize the transaction within this realm.

Step 6 Join the appliance to the domain:

a) Configure the Active Directory Account:

Setting	Description
Active Directory Domain	The Active Directory server domain name. Also known as a DNS Domain or realm.

Setting	Description
NetBIOS domain name	If the network uses NetBIOS, provide the domain name. **Tip** If this option is not available use the `setntlmsecuritymode` CLI command to verify that the NTLM security mode is set to "domain."
Computer Account	Specify a location within the Active Directory domain where AsyncOS will create an Active Directory computer account, also known as a "machine trust account," to uniquely identify the computer on the domain. If the Active Directory environment automatically deletes computer objects at particular intervals, specify a location for the computer account that is in a container, protected from automatic deletion.
Enable Trusted Domain Lookup	**Enable Trusted Domain Lookup** option is added in the **Active Directory Account** section (**Network** > **Authentication** > **Add Realm**) to control the behavior of the trusted domain lookup for the realm. The option is enabled by default.

b) Click **Join Domain**.

Note
If you attempt to join a domain you have already joined (even if you use the same credentials), existing connections will be closed, as the Active Directory will send a new set of keys to all clients including this Secure Web Appliance. Affected clients will need to log off and log back in again.

Note
The hostname of the Secure Web Appliance deployed on AWS must be unique. You must modify the first string of the hostname to create a unique hostname.

For example, if "mgmt" is appended to the hostname as the first string, you can modify it as "mgmt<*wsa_hostname*>".

c) Provide login credentials (user name and passphrase) for the account on the Active Directory, and click Create Account.

Step 7 (Optional) Configure transparent user identification.

Setting	Description
Enable Transparent User Identification using Active Directory agent	Enter both the server name for the machine where the primary Context Directory agent is installed and the shared secret used to access it. (Optional) Enter the server name for the machine where a backup Context Directory agent is installed and its shared secret.

Step 8 Configure Network Security:

Setting	Description
Client Signing Required	Select this option if the Active Directory server is configured to require client signing. The selection of this option enables SMB signing to: • Place the digital signature when the appliance connects to the Active Directory. • Prevent man-in-the-middle attacks.

Step 9 If you will use high availability, check the **Use keytab authentication** check box in the Kerberos High Availability section.

a) Enter the Username and Password.

Enter the username of Active Directory user name associated with SPN(s) corresponding to the IP address or hostname of the high availability cluster. Do not include the domain name with the user name (for example, enter 'johndoe', rather than 'DOMAIN\johndoe', or 'johndoe@domain'). See Creating a Service Account in Windows Active Directory for Kerberos Authentication in High Availability Deployments, on page 11 for specific information about creating a service account that will be used for authentication in high availability deployments.

b) Repeat this step for all appliances in the high availability cluster.

Note
If your appliance resides behind a HTTP/HTTPS traffic distribution device like a load balancer, you should associate the SPN of the traffic distribution device in the Active Directory with a user account, and enter the credentials of that user account in the Kerberos High Availability section. The SPN of the first device that redirects traffic in the network topology should be added. For example, if client devices' outbound network traffic passes through a traffic manager, a load balancer, and then to the Secure Web Appliance, the SPN for the traffic manager should be added to a user account on the Active Directory, and the user credentials should be entered in this section. This is because the traffic manager is the first device that encounters client devices' traffic.

Step 10 (Optional) Click **Start Test.** This will test the settings you have entered, ensuring they are correct before real users use them to authenticate. For details on the testing performed, see Using Multiple NTLM Realms and Domains, on page 26.

Step 11 Troubleshoot any issues found during testing. See Troubleshooting Tools for Authentication Issues.

Step 12 Submit and commit your changes.

What to do next

Create an Identification Profile that uses the Kerberos authentication scheme. Classifying Users and Client Software.

How to Create an Active Directory Authentication Realm (NTLMSSP and Basic)

Prerequisites for Creating an Active Directory Authentication Realm (NTLMSSP and Basic)

- Ensure you have the rights and domain information needed to join the Secure Web Appliance to the Active Directory domain you wish to authenticate against.

- If you plan to use “domain” as the NTLM security mode, use only nested Active Directory groups. If Active Directory groups are not nested, use the default value, “ads”. See `setntlmsecuritymode` in the Command Line Interface topic of this guide.
- Compare the current time on the Secure Web Appliance with the current time on the Active Directory server and verify that the difference is no greater than the time specified in the “Maximum tolerance for computer clock synchronization” option on the Active Directory server.
- If the Secure Web Appliance is managed by a Security Management appliance, be prepared to ensure that same-named authentication realms on different Secure Web Appliances have identical properties defined on each appliance.
- Be aware that once you commit the new realm, you cannot change a realm’s authentication protocol.
- The Secure Web Appliance needs to connect to the domain controllers for all trusted domains, and to the configured domain controllers into the NTLM realm. For authentication to work correctly, you need to open the following ports to all domain controllers on the internal domain and on the external domain:
  - LDAP (389 UDP and TCP)
  - Microsoft SMB (445 TCP)
  - Kerberos (88 TCP)
  - End-point resolution – port mapper (135 TCP) Net Log-on fixed port
- For NTLMSSP, single sign on (SSO) can be configured on client browsers. See Configuring Single-Sign-on, on page 10.

About Using Multiple NTLM Realms and Domains

The following rules apply in regard to using multiple NTLM realms and domains:

- You can create up to 10 NTLM authentication realms.
- The client IP addresses in one NTLM realm must not overlap with the client IP addresses in another NTLM realm.
- Each NTLM realm can join one Active Directory domain only but can authenticate users from any domains trusted by that domain. This trust applies to other domains in the same forest by default and to domains outside the forest to which at least a one way trust exists.
- Create additional NTLM realms to authenticate users in domains that are not trusted by existing NTLM realms.

Creating an Active Directory Authentication Realm (NTLMSSP and Basic)

Before you begin

Ensure that the higher range ports in the appliance (49152-65535) are unblocked in your firewall. These ports are required to perform the asynchronous group lookup requests. Blocking these ports may cause intermittent authentication failure.

Procedure

Step 1 Choose **Network** > **Authentication**.

Step 2 Click **Add Realm**.

Step 3 Assign a unique name to the authentication realm using only alphanumeric and space characters.

Step 4 Select **Active Directory** in the Authentication Protocol and Scheme(s) field.

Step 5 Enter up to three fully-qualified domain names or IP addresses for the Active Directory server(s).

Example: `active.example.com` .

An IP address is required only if the DNS servers configured on the appliance cannot resolve the Active Directory server hostname.

When multiple authentication servers are configured in the realm, the appliance attempts to authorize with up to three authentication servers before failing to authorize the transaction within this realm.

Step 6 Join the appliance to the domain:

a) Configure the Active Directory Account:

Setting	Description
Active Directory Domain	The Active Directory server domain name. Also known as a DNS Domain or realm.
NetBIOS domain name	If the network uses NetBIOS, provide the domain name.
Computer Account	Specify a location within the Active Directory domain where AsyncOS will create an Active Directory computer account, also known as a "machine trust account", to uniquely identify the computer on the domain. If the Active Directory environment automatically deletes computer objects at particular intervals, specify a location for the computer account that is in a container, protected from automatic deletion.
Enable Trusted Domain Lookup	**Enable Trusted Domain Lookup** option is added in the **Active Directory Account** section (**Network** > **Authentication** > **Add Realm**) to control the behavior of the trusted domain lookup for the realm. The option is enabled by default.

b) Click **Join Domain**.

Note
If you attempt to join a domain you have already joined (even if you use the same credentials), existing connections will be closed, as the Active Directory will send a new set of keys to all clients including this Secure Web Appliance. Affected clients will need to log off and log back in again.

Note
The hostname of the Secure Web Appliance deployed on AWS must be unique. You must modify the first string of the hostname to create a unique hostname.

For example, if "mgmt" is appended to the hostname as the first string, you can modify it as "mgmt<*wsa_hostname*>".

c) Enter the sAMAccountName user name and passphrase for an existing Active Directory user that has rights to create computer accounts in the domain.

Example: `"jazzdoe"` Do not use: `"DOMAIN\jazzdoe"` or `"jazzdoe@domain"`

This information is used once to establish the computer account and is not saved.

d) Click **Create Account**.

Step 7 (Optional) Configure transparent authentication.

Setting	Description
Enable Transparent User Identification using Active Directory agent	Enter both the server name for the machine where the primary Context Directory agent is installed and the shared secret used to access it. (Optional) Enter the server name for the machine where a backup Context Directory agent is installed and its shared secret.

Step 8 Configure Network Security:

Setting	Description
Client Signing Required	Select this option if the Active Directory server is configured to require client signing. The selection of this option enables SMB signing to: • Place the digital signature when the appliance connects to the Active Directory. • Prevent man-in-the-middle attacks.

Step 9 (Optional) Click **Start Test.** This will test the settings you have entered, ensuring they are correct before real users use them to authenticate.

Step 10 Submit and commit your changes.

Creating an LDAP Authentication Realm

Before you begin

- Obtain the following information about LDAP in your organization:
  - LDAP version
  - Server addresses
  - LDAP ports
- If the Secure Web Appliance is managed by a Security Management appliance, ensure that same-named authentication realms on different Secure Web Appliances have identical properties defined on each appliance.

Procedure

Step 1 Choose **Network > Authentication**.

Step 2 Click **Add Realm**.

Step 3 Assign a unique name to the authentication realm using only alphanumeric and space characters.

Step 4 Select **LDAP** in the Authentication Protocol and Scheme(s) field.

Step 5 Enter the LDAP authentication settings:

Setting	Description
LDAP Version	Choose the version of LDAP, and choose whether or not to use Secure LDAP. The appliance supports LDAP versions 2 and 3. Secure LDAP requires LDAP version 3. Choose whether or not this LDAP server supports Novell eDirectory to use with transparent user identification.
LDAP Server	Enter the LDAP server IP address or hostname and its port number. You can specify up to three servers. The hostname must be a fully-qualified domain name. For example, `ldap.example.com`. An IP address is required only if the DNS servers configured on the appliance cannot resolve the LDAP server hostname. The default port number for Standard LDAP is 389. The default number for Secure LDAP is 636. If the LDAP server is an Active Directory server, enter the hostname or IP address and the port of the domain controller here. Whenever possible, enter the name of the Global Catalog Server and use port 3268. However, you might want to use a local domain controller when the global catalog server is physically far away and you know you only need to authenticate users on the local domain controller. **Note:** When you configure multiple authentication servers in the realm, the appliance attempts to authorize with up to three authentication servers before failing to authenticate the transaction within that realm. From AsyncOS version 11.5 onwards, you can specify the source interface for LDAP/NTLM (Domain Controller communication). Select the **Set Source Interface** check box, and then select the Source Interface from the drop-down.
LDAP Persistent Connections (under the Advanced section)	Choose one of the following values: • **Use persistent connections (unlimited)**. Use existing connections. If no connections are available a new connection is opened. • **Use persistent connections.** Use existing connections to service the number of requests specified. When the maximum is reached, establish a new connection to the LDAP server. • **Do not use persistent connections.** Always create a new connection to the LDAP server.

Setting	Description
User Authentication	Enter values for the following fields: **Base Distinguished Name (Base DN)** The LDAP database is a tree-type directory structure and the appliance uses the Base DN to navigate to the correct location in the LDAP directory tree to begin a search. A valid Base DN filter string is composed of one or more components of the form `object-value`. For example `dc=companyname, dc=com`. **Note** After you upgrade to this release, you cannot perform the Start Test for LDAP authentication if this field is empty. **User Name Attribute** Choose one of the following values: • **uid**, **cn**, and **sAMAccountName.** Unique identifiers in the LDAP directory that specify a username. • **custom.** A custom identifier such as `UserAccount`. **User Filter Query** The User Filter Query is an LDAP search filter that locates the users Base DN. This is required if the user directory is in a hierarchy below the Base DN, or if the login name is not included in the user-specific component of that users Base DN. Choose one of the following values: • **none.** Filters any user. • **custom.** Filters a particular group of users.
Query Credentials	Choose whether or not the authentication server accepts anonymous queries. If the authentication server does accept anonymous queries, choose **Server Accepts Anonymous Queries**. If the authentication server does not accept anonymous queries, choose **Use Bind DN** and then enter the following information: • **Bind DN.** The user on the external LDAP server permitted to search the LDAP directory. Typically, the bind DN should be permitted to search the entire directory. • **Passphrase.** The passphrase associated with the user you enter in the Bind DN field. The following text lists some example users for the Bind DN field: cn=administrator,cn=Users,dc=domain,dc=com sAMAccountName=jdoe,cn=Users,dc=domain,dc=com. If the LDAP server is an Active Directory server, you may also enter the Bind DN username as "DOMAIN\username."

Step 6 (Optional) Enable Group Authorization via group object or user object and complete the settings for the chosen option accordingly:

Group Object Setting	Description
Group Membership Attribute Within Group Object	Choose the LDAP attribute which lists all users that belong to this group. Choose one of the following values: • **member** and **uniquemember.** Unique identifiers in the LDAP directory that specify group members. • **custom.** A custom identifier such as `UserInGroup`.
Attribute that Contains the Group Name	Choose the LDAP attribute which specifies the group name that can be used in the policy group configuration. Choose one of the following values: • **cn.** A unique identifier in the LDAP directory that specifies the name of a group. • **custom.** A custom identifier such as `FinanceGroup`.
Query String to Determine if Object is a Group	Choose an LDAP search filter that determines if an LDAP object represents a user group. Choose one of the following values: • **objectclass=groupofnames** • **objectclass=groupofuniquenames** • **objectclass=group** • **custom.** A custom filter such as `objectclass=person`. **Note:** The query defines the set of authentication groups which can be used in policy groups.

User Object Setting	Description
Group Membership Attribute Within User Object	Choose the attribute which list all the groups that this user belongs to. Choose one of the following values: • **memberOf.** Unique identifiers in the LDAP directory that specify user members. • **custom.** A custom identifier such as `UserInGroup`.
Group Membership Attribute is a DN	Specify whether the group membership attribute is a distinguished name (DN) which refers to an LDAP object. For Active Directory servers, enable this option. When this is enabled, you must configure the subsequent settings.
Attribute that Contains the Group Name	When the group membership attribute is a DN, this specifies the attribute that can be used as group name in policy group configurations. Choose one of the following values: • **cn.** A unique identifier in the LDAP directory that specifies the name of a group. • **custom.** A custom identifier such as `FinanceGroup`.

User Object Setting	Description
Query String to Determine if Object is a Group	Choose an LDAP search filter that determines if an LDAP object represents a user group. Choose one of the following values: • **objectclass=groupofnames** • **objectclass=groupofuniquenames** • **objectclass=group** • **custom.** A custom filter such as `objectclass=person`. **Note:** The query defines the set of authentication groups which can be used in Web Security Manager policies.

Step 7 (Optional) Configure external LDAP authentication for users.

a) Select **External Authentication Queries**.

b) Identify the user accounts:.

Base DN	The Base DN to navigate to the correct location in the LDAP directory tree to begin a search.

Query String	The query to return the set of authentication groups, for example:
	`(&(objectClass=posixAccount)(uid={u}))`
	or
	`(&(objectClass=user)(sAMAccountName={u}))`

Attribute containing the user's full name	The LDAP attribute, for example, displayName or gecos .

c) (Optional) Deny login to expired accounts based on RFC 2307 account expiration LDAP attributes.

d) Provide a query to retrieve group information for users.

If a user belongs to multiple LDAP groups with different user roles, AsyncOS grants the user the permissions for the most restrictive role.

Base DN	The Base DN to navigate to the correct location in the LDAP directory tree to begin a search.
Query String	`(&(objectClass=posixAccount)(uid={u}))`
Attribute containing the user's full name	`gecos`

Step 8 (Optional) Click **Start Test.** This will test the settings you have entered, ensuring they are correct before real users use them to authenticate. For details on the testing performed, see Using Multiple NTLM Realms and Domains, on page 26.

Note
Once you submit and commit your changes, you cannot later change a realm's authentication protocol.

Step 9 Submit and commit your changes.

What to do next

Create an Identification Profile that uses the Kerberos authentication scheme. See Classifying Users and Client Software.

Related Topics

- External Authentication, on page 14

Using Multiple NTLM Realms and Domains

The following rules apply in regard to using multiple NTLM realms and domains:

- You can create up to 10 NTLM authentication realms.
- The client IP addresses in one NTLM realm must not overlap with the client IP addresses in another NTLM realm.
- Each NTLM realm can join one Active Directory domain only but can authenticate users from any domains trusted by that domain. This trust applies to other domains in the same forest by default and to domains outside the forest to which at least a one way trust exists.
- Create additional NTLM realms to authenticate users in domains that are not trusted by existing NTLM realms.

About Deleting Authentication Realms

Deleting an authentication realm disables associated identities, which in turn removes those identities from associated policies.

Deleting an authentication realm removes it from sequences.

Configuring Global Authentication Settings

Configure Global Authentication Settings to apply settings to all authentication realms, independent of their authentication protocols.

The Web Proxy deployment mode affects which global authentication settings you can configure. More settings are available when it is deployed in transparent mode than in explicit forward mode.

Before you begin

Be familiar with the following concepts:

- Failed Authentication, on page 35
- Failed Authorization: Allowing Re-Authentication with Different Credentials, on page 39

Procedure

Step 1 Choose **Network** > **Authentication**

Step 2 Click **Edit Global Settings**.

Step 3 Edit the settings in the Global Authentication Settings section:.

Setting	Description
Action if Authentication Service Unavailable	Choose one of the following values: • **Permit traffic to proceed without authentication.** Processing continues as if the user was authenticated. • **Block all traffic if user authentication fails.** Processing is discontinued and all traffic is blocked.
Failed Authentication Handling	When you grant users guest access in an Identification Profile policy, this setting determines how the Web Proxy identifies and logs the user as a guest in the access logs. For more information on granting users guest access, see Granting Guest Access After Failed Authentication, on page 38.
Re-authentication (Enable Re-Authentication Prompt If End User Blocked by URL Category or User Session Restriction)	This setting allows users to authenticate again if the user is blocked from a website due to a restrictive URL filtering policy or due to being restricted from logging into another IP address. The user sees a block page that includes a link that allows them to enter new authentication credentials. If the user enters credentials that allow greater access, the requested page appears in the browser. **Note:** This setting only applies to authenticated users who are blocked due to restrictive URL filtering policies or User Session Restrictions. It does not apply to blocked transactions by subnet with no authentication. For more information, see Failed Authorization: Allowing Re-Authentication with Different Credentials, on page 39.
Basic Authentication Token TTL	Controls the length of time that user credentials are stored in the cache before revalidating them with the authentication server. This includes the username and passphrase and the directory groups associated with the user. The default value is the recommended setting. When the Surrogate Timeout setting is configured and is greater than the Basic Authentication Token TTL, then the Surrogate Timeout value takes precedence and the Web Proxy contacts the authentication server after surrogate timeout expires.

The remaining authentication settings you can configure depends on how the Web Proxy is deployed, in transparent or explicit forward mode.

Step 4 If the Web Proxy is deployed in transparent mode, edit the settings as follows:

Setting	Description
Credential Encryption	This setting specifies whether or not the client sends the login credentials to the Web Proxy through an encrypted HTTPS connection. This setting applies to both Basic and NTLMSSP authentication schemes, but it is particularly useful for Basic authentication scheme because user credentials are sent as plain text. For more information, see Failed Authentication, on page 35.
HTTPS Redirect Port	Specify a TCP port to use for redirecting requests for authenticating users over an HTTPS connection. This specifies through which port the client will open a connection to the Web Proxy using HTTPS. This occurs when credential encryption is enabled or when using Access Control and users are prompted to authenticate.
Redirect Hostname	Enter the short hostname of the network interface on which the Web Proxy listens for incoming connections. When you configure authentication on an appliance deployed in transparent mode, the Web Proxy uses this hostname in the redirection URL sent to clients for authenticating users. You can enter either the following values: • **Single word hostname.** You can enter the single word hostname that is DNS resolvable by the client and the Secure Web Appliance. This allows clients to achieve true single sign-on with Internet Explorer without additional browser side setup. Be sure to enter the single word hostname that is DNS resolvable by the client and the Secure Web Appliance. For example, if your clients are in domain mycompany.com and the interface on which the Web Proxy is listening has a full hostname of **`proxy.mycompany.com`**, then you should enter **`proxy`** in this field. Clients perform a lookup on proxy and they should be able to resolve **`proxy.mycompany.com`**. • **Fully qualified domain name (FQDN).** You can also enter the FQDN or IP address in this field. However, if you do that and want true single sign-on for Internet Explorer and Firefox browsers, you must ensure that the FQDN or IP address is added to the client's Trusted Sites list in the client browsers. The default value is the FQDN of the M1 or P1 interface, depending on which interface is used for proxy traffic.
Credential Cache Options: Surrogate Timeout	This setting specifies how long the Web Proxy waits before asking the client for authentication credentials again. Until the Web Proxy asks for credentials again, it uses the value stored in the surrogate (IP address or cookie). It is common for user agents, such as browsers, to cache the authentication credentials so the user will not be prompted to enter credentials each time.

Setting	Description
Credential Cache Options: Client IP Idle Timeout	When IP address is used as the authentication surrogate, this setting specifies how long the Web Proxy waits before asking the client for authentication credentials again when the client has been idle. When this value is greater than the Surrogate Timeout value, this setting has no effect and clients are prompted for authentication after the Surrogate Timeout is reached. You might want to use this setting to reduce the vulnerability of users who leave their computers.
User Session Restrictions	This setting specifies whether or not authenticated users are allowed to access the Internet from multiple IP addresses simultaneously. You might want to restrict access to one machine to prevent users from sharing their authentication credentials with non-authorized users. When a user is prevented from logging in at a different machine, an end-user notification page appears. You can choose whether or not users can click a button to login as a different username using the Re-authentication setting on this page. When you enable this setting, enter the restriction timeout value, which determines how long users must wait before being able to log into a machine with a different IP address. The restriction timeout value must be greater than the surrogate timeout value. You can remove a specific user or all users from the authentication cache using the `authcache` CLI command.

Setting	Description
Header Based Authentication	This setting enables you to configure the Header Based Authentication scheme for an active directory. The cache settings for Header Based Authentication: • Authentication cache is enabled by default. • Authentication cache timeout is the same as that of surrogate timeout. • Cache stores the username and the user groups. **Note** Clear the authentication cache if you update the User Group configuration. Check the **Standard Header** check box with ASCII as text encoding and No encoding for Binary which are the default settings. Enable the **Use Groups in X-Authenticate-Groups Header/Custom Header for matching Access Policies** check box for considering the incoming groups header. Use **Custom Header Name** option if you want to configure the custom header names. **Note** If you select the **Use Groups in X-Authenticate-Groups Header/Custom Header for matching Access Policies** check box, and no X-Authenticated-Groups header is provided, then the match may fail for access policies. If it is not enabled, then the Groups that are fetched from the active directory will be matched against the access policies. Enable the **Retain Authentication Details on Egress** check box to retain the headers (user and groups headers) on the egress.
Advanced	When using Credential Encryption or Access Control, you can choose whether the appliance uses the digital certificate and key shipped with the appliance (the Cisco Web Security Appliance Demo Certificate) or a digital certificate and key you upload here.

Step 5 If the Web Proxy is deployed in explicit forward mode, edit the settings as follows:

Setting	Description
Credential Encryption	This setting specifies whether or not the client sends the login credentials to the Web Proxy through an encrypted HTTPS connection. To enable credential encryption, choose “HTTPS Redirect (Secure)”. When you enable credential encryption, additional fields appear to configure how to redirect clients to the Web Proxy for authentication. This setting applies to both Basic and NTLMSSP authentication schemes, but it is particularly useful for Basic authentication scheme because user credentials are sent as plain text. For more information, see Failed Authentication, on page 35.

Setting	Description
HTTPS Redirect Port	Specify a TCP port to use for redirecting requests for authenticating users over an HTTPS connection. This specifies through which port the client will open a connection to the Web Proxy using HTTPS. This occurs when credential encryption is enabled or when using Access Control and users are prompted to authenticate.
Redirect Hostname	Enter the short host name of the network interface on which the Web Proxy listens for incoming connections. When you enable Authentication Mode above, the Web Proxy uses this hostname in the redirection URL sent to clients for authenticating users. You can enter either the following values: • **Single word hostname.** You can enter the single word host name that is DNS resolvable by the client and the Secure Web Appliance. This allows clients to achieve true single sign-on with Internet Explorer without additional browser side setup. Be sure to enter the single word host name that is DNS resolvable by the client and the Secure Web Appliance. For example, if your clients are in domain mycompany.com and the interface on which the Web Proxy is listening has a full host name of `proxy.mycompany.com`, then you should enter `proxy` in this field. Clients perform a lookup on proxy and they should be able to resolve `proxy.mycompany.com`. • **Fully qualified domain name (FQDN).** You can also enter the FQDN or IP address in this field. However, if you do that and want true single sign-on for Internet Explorer and Firefox browsers, you must ensure that the FQDN or IP address is added to the client's Trusted Sites list in the client browsers. The default value is the FQDN of the M1 or P1 interface, depending on which interface is used for proxy traffic.
Credential Cache Options: Surrogate Timeout	This setting specifies how long the Web Proxy waits before asking the client for authentication credentials again. Until the Web Proxy asks for credentials again, it uses the value stored in the surrogate (IP address or cookie). Note that it is common for user agents, such as browsers, to cache the authentication credentials so the user will not be prompted to enter credentials each time.
Credential Cache Options: Client IP Idle Timeout	When IP address is used as the authentication surrogate, this setting specifies how long the Web Proxy waits before asking the client for authentication credentials again when the client has been idle. When this value is greater than the Surrogate Timeout value, this setting has no effect and clients are prompted for authentication after the Surrogate Timeout is reached. You might want to use this setting to reduce the vulnerability of users who leave their computers.

Setting	Description
User Session Restrictions	This setting specifies whether or not authenticated users are allowed to access the Internet from multiple IP addresses simultaneously. You might want to restrict access to one machine to prevent users from sharing their authentication credentials with non-authorized users. When a user is prevented from logging at a different machine, an end-user notification page appears. You can choose whether or not users can click a button to login as a different username using the Re-authentication setting on this page. When you enable this setting, enter the restriction timeout value, which determines how long users must wait before being able to log into a machine with a different IP address. The restriction timeout value must be greater than the surrogate timeout value. You can remove a specific user or all users from the authentication cache using the `authcache` CLI command.
Header Based Authentication	This setting enables you to configure the Header Based Authentication scheme for an active directory. The cache settings for Header Based Authentication: • Authentication cache is enabled by default. • Authentication cache timeout is the same as that of surrogate timeout. • Cache stores the username and the user groups. **Note** Clear the authentication cache if you update the User Group configuration. Check the **Standard Header** check box with ASCII as text encoding and No encoding for Binary which are the default settings. Enable the **Use Groups in X-Authenticate-Groups Header/Custom Header for matching Access Policies** check box for considering the incoming groups header. Use **Custom Header Name** option if you want to configure the custom header names. **Note** If you select the **Use Groups in X-Authenticate-Groups Header/Custom Header for matching Access Policies** check box, and no X-Authenticated-Groups header is provided, then the match may fail for access policies. If it is not enabled, then the Groups that are fetched from the active directory will be matched against the access policies. Enable the **Retain Authentication Details on Egress** check box to retain the headers (user and groups headers) on the egress.

Setting	Description
Advanced	When using Credential Encryption or Access Control, you can choose whether the appliance uses the digital certificate and key shipped with the appliance (the Cisco Web Security Appliance Demo Certificate) or a digital certificate and key you upload here. To upload a digital certificate and key, click **Browse** and navigate to the necessary file on your local machine. Then click **Upload Files** after you select the files you want.

Step 6 Submit and commit your changes.

Authentication Sequences

- About Authentication Sequences, on page 33
- Creating Authentication Sequences, on page 34
- Editing And Reordering Authentication Sequences, on page 34
- Deleting Authentication Sequences, on page 35

About Authentication Sequences

Use authentication sequences to allow single Identities to authenticate users via different authentication servers or protocols. Authentication sequences are also useful for providing backup options in case primary authentication options become unavailable.

Authentication sequences are collections of two or more authentication realms. The realms used can have different authentication servers and different authentication protocols. For more information on authentication realms, see Authentication Realms, on page 13.

After you create a second authentication realm, the appliance automatically displays a Realm Sequences section under Network > Authentication and includes a default authentication sequence named All Realms. The All Realms sequence automatically includes each realm you define. You can change the order of the realms within the All Realms sequence, but you cannot delete the All Realms sequence or remove any realms from it.

When multiple NTLM authentication realms are defined, the Secure Web Appliance uses the NTLMSSP authentication scheme with only one NTLM authentication realm per sequence. You can choose which NTLM authentication realm to use for NTLMSSP within each sequence, including the All Realms sequence. To use NTLMSSP with multiple NTLM realms, configure a single Identification Profile for two authentication Realms ensuring that one identity is used for All Realms. The Realms must have mutual trust between them.

Which authentication realms within a sequence get used during authentication depends on:

- The authentication scheme used. This is generally dictated by the type of credentials entered at the client.
- The order in which realms are listed within the sequence (for Basic realms only, as only one NTLMSSP realm is possible).

Tip For optimal performance, authenticate clients on the same subnet using a single realm.

Creating Authentication Sequences

Before you begin

- Create two or more authentication realms (see Authentication Realms, on page 13).
- If the Secure Web Appliance is managed by a Security Management appliance, ensure that same-named authentication realms on different Secure Web Appliances have identical properties defined on each appliance.
- Be aware that AsyncOS will use the realms to process authentication sequentially, beginning with the first realm in the list.

Procedure

Step 1 Choose **Network > Authentication**

Step 2 Click **Add Sequence**.

Step 3 Enter a unique name for the sequence using alphanumeric and space characters.

Step 4 In the first row of the Realm Sequence for Basic Scheme area, choose the first authentication realm you want to include in the sequence.

Step 5 In the second row of the Realm Sequence for Basic Scheme area, choose the next realm you want to include in the sequence.

Step 6 (Optional) Click **Add Row** to include another realm that uses Basic credentials.

Step 7 If an NTLM realm is defined, choose an NTLM realm in the Realm for NTLMSSP Scheme field.

The Web Proxy uses this NTLM realm when the client sends NTLMSSP authentication credentials.

Step 8 Submit and commit your changes.

Editing And Reordering Authentication Sequences

Procedure

Step 1 Choose **Network > Authentication**.

Step 2 Click the name of the sequence you wish to edit or re-order.

Step 3 Choose a realm name from the Realms drop-down list on the row corresponding to the position number you want the realm to occupy in the sequence.

Note

For the All Realms sequence, you can only change the order of its realms, you cannot change the realms themselves. To change the order of realms in the All Realms sequence, click the arrows in the Order column to reposition the corresponding realms.

Step 4 Repeat **Step 3** until all realms are listed and ordered as required, ensuring that each realm name appears in one row only.

Step 5 Submit and commit your changes.

Deleting Authentication Sequences

Before you begin

Be aware that deleting an authentication sequence also disables associated identities, which in turn removes those identities from associated policies.

Procedure

Step 1 Choose **Network** > **Authentication**.

Step 2 Click the trash can icon for the sequence name.

Step 3 Click **Delete to** confirm that you want to delete the sequence.

Step 4 Commit your changes.

Failed Authentication

- About Failed Authentication, on page 35
- Bypassing Authentication with Problematic User Agents , on page 36
- Bypassing Authentication, on page 37
- Permitting Unauthenticated Traffic While Authentication Service is Unavailable, on page 37
- Granting Guest Access After Failed Authentication, on page 38
- Failed Authorization: Allowing Re-Authentication with Different Credentials, on page 39

About Failed Authentication

Users may be blocked from the web due to authentication failure for the following reasons:

- **Client/user agent limitations.** Some client applications may not properly support authentication. You can bypass authentication for these clients by configuring Identification Profiles that do not require authorization and basing their criteria on the clients (and, optionally, on the URLs they need to access).
- **Authentication service is unavailable.** An authentication service might be unavailable due to network or server issues. You can choose to allow unauthenticated traffic in this circumstance.
- **Invalid credentials.** Some users may be unable to supply valid credentials for proper authentication (for example, visitors or users awaiting credentials). You can choose to grant these users limited access to the web.

Related Topics

- Bypassing Authentication with Problematic User Agents , on page 36
- Bypassing Authentication, on page 37
- Permitting Unauthenticated Traffic While Authentication Service is Unavailable, on page 37
- Granting Guest Access After Failed Authentication, on page 38

Bypassing Authentication with Problematic User Agents

Some user agents are known to have authentication issues that can impact normal operations.

You should bypass authentication via the following user agents:

- Windows-Update-Agent
- MICROSOFT_DEVICE_METADATA_RETRIEVAL_CLIENT
- Microsoft BITS
- SLSSoapClient
- Akamai NetSession Interface
- Microsoft-CryptoAPI
- NCSI
- MSDW
- Gnotify
- msde
- Google Update



Note The access policies will still filter (based on URL categories) and scan (McAfee, Webroot) traffic as per the access policy setup.

Procedure

Step 1 Configure the Identification Profile to bypass authentication with the specified user agents:

a) Select **Web Security Manager** > **Identification Profile**.
b) Click **Add Identification Profile**.
c) Enter information:

Option	Value
Name	User Agent AuthExempt Identification Profile
Insert Above	Set to the first profile in the processing order
Define Members by Subnet	Leave blank.
Define Members by Authentication	No Authentication Required.

d) Click **Advanced** > **User Agents**.
e) Click **None Selected**.

f) Under Custom user Agents, specify the problematic User Agent strings.

Step 2 Configure the Access Policy:

a) Choose **Web Security Manager** > **Access Policies**.
b) Click **Add Policy**.
c) Enter information:

Option	Value
Policy Name	Auth Exemption for User Agents
Insert Above Policy	Set to the first policy in the processing order.
Identification Profile Policy	User Agent AuthExempt Identification Profile
Advanced	None

Step 3 Submit and commit your changes.

Bypassing Authentication

	Step	More Information
1	Create a custom URL category that contains the affected websites by configuring the Advanced properties.	Creating and Editing Custom URL Categories
2	Create an Identification Profile with these characteristics: • Placed above all identities that require authentication. • Includes the custom URL category. • Includes affected client applications. • Does not require authentication	Classifying Users and Client Software
3	Create a policy for the Identification Profile.	Creating a Policy

Related Topics

- Bypassing the Web Proxy

PermittingUnauthenticatedTrafficWhileAuthenticationServiceisUnavailable

Note This configuration applies only when an authentication service is unavailable. It will not bypass authentication permanently. For alternative options, see About Failed Authentication, on page 35

Procedure

Step 1 Choose **Network** > **Authentication**.

Step 2 Click **Edit Global Settings**.

Step 3 Click the **Permit Traffic To Proceed Without Authentication** in the Action If Authentication Service Unavailable field.

Step 4 Submit and commit your changes.

Granting Guest Access After Failed Authentication

Granting guest access requires that the following procedures are completed:

1. Define an Identification Profile that Supports Guest Access, on page 38
2. Use an Identification Profile that Supports Guest Access in a Policy, on page 38
3. (Optional) Configure How Guest User Details are Logged, on page 39



Note If an Identification Profile allows guest access and there is no user-defined policy that uses that Identification Profile, users who fail authentication match the global policy of the applicable policy type. For example, if MyIdentificationProfile allows guest access and there is no user-defined Access Policy that uses MyIdentificationProfile, users who fail authentication match the global Access Policy. If you do not want guest users to match a global policy, create a policy above the global policy that applies to guest users and blocks all access.

Define an Identification Profile that Supports Guest Access

Procedure

Step 1 Choose **Web Security Manager** > **Identification Profiles**.

Step 2 Click **Add Identification Profile** to add a new identity, or click the name of an existing identity that you wish to use.

Step 3 Check the **Support Guest Privileges** check box.

Step 4 Submit and commit your changes.

Use an Identification Profile that Supports Guest Access in a Policy

Procedure

Step 1 Choose a policy type from the Web Security Manager menu.

Step 2 Click a policy name in the policies table.

Step 3 Choose **Select One Or More Identification Profiles** from the Identification Profiles And Users drop-down list (if not already chosen).

Step 4 Choose a **profile** that supports guest access from the drop-down list in the Identification Profile column.

Step 5 Click the **Guests (Users Failing Authentication)** radio button.

Note
If this option is not available it means the **profile** you chose is not configured to support guest access. Return to step 4 and choose another, or see Define an Identification Profile that Supports Guest Access, on page 38 to define a new one.

Step 6 Submit and commit your changes.

Configure How Guest User Details are Logged

Procedure

Step 1 Choose **Network > Authentication.**

Step 2 Click **Edit Global Settings**.

Step 3 Click a Log Guest User By radio button, described below, in the Failed Authentication Handling field.

Radio button	Description
IP Address	The IP address of the guest user's client will be logged in the access logs.
User Name As Entered By End-User	The user name that originally failed authentication will be logged in the access logs.

Step 4 Submit and commit your changes.

Failed Authorization: Allowing Re-Authentication with Different Credentials

- About Allowing Re-Authentication with Different Credentials, on page 39
- Allowing Re-Authentication with Different Credentials, on page 40

About Allowing Re-Authentication with Different Credentials

Use re-authentication to allow users the opportunity to authenticate again, using different credentials, if the credentials they previously used have failed authorization. A user may authenticate successfully but still be prevented from accessing a web resource if not authorized to do so. This is because authentication merely identifies users for the purpose of passing their verified credentials on to policies, but it is the policies that authorize those users (or not) to access resources.

A user must have authenticated successfully to be allowed to re-authenticate.

- To use the re-authentication feature with user defined end-user notification pages, the CGI script that parses the redirect URL must parse and use the Reauth_URL parameter.

Allowing Re-Authentication with Different Credentials

Procedure

Step 1 Choose **Network** > **Authentication**.

Step 2 Click **Edit Global Settings.**

Step 3 Check the **Re-Authentication Prompt If End User Blocked by URL Category Or User Session Restriction** check box.

Step 4 Click **Submit**.

Tracking Identified Users

Note When the appliance is configured to use cookie-based authentication surrogates, it does not get cookie information from clients for HTTPS and FTP over HTTP requests. Therefore, it cannot get the user name from the cookie.

Supported Authentication Surrogates for Explicit Requests

Surrogate Types	Credential Encryption Disabled			Credential Encryption Enabled		
Protocol:	HTTP	HTTPS & FTP over HTTP	Native FTP	HTTP	HTTPS & FTP over HTTP	Native FTP
No Surrogate	Yes	Yes	Yes	NA	NA	NA
IP-based	Yes	Yes	Yes	Yes	Yes	Yes
Cookie-based	Yes	Yes***	Yes***	Yes	No/Yes**	Yes***

Supported Authentication Surrogates for Transparent Requests

Note See also the description of the Authentication Surrogates options in Classifying Users and Client Software.

Surrogate Types	Credential Encryption Disabled			Credential Encryption Enabled		
Protocol:	HTTP	HTTPS	Native FTP	HTTP	HTTPS	Native FTP
No Surrogate	NA	NA	NA	NA	NA	NA

Surrogate Types	Credential Encryption Disabled			Credential Encryption Enabled		
IP-based	Yes	No/Yes*	No/Yes*	Yes	No/Yes*	No/Yes*
Cookie-based	Yes	No/Yes**	No/Yes**	Yes	No/Yes**	No/Yes**

* Works after the client makes a request to an HTTP site and is authenticated. Before this happens, the behavior depends on the transaction type:

- **Native FTP transactions.** Transactions bypass authentication.
- **HTTPS transactions.** Transactions are dropped. However, you can configure the HTTPS Proxy to decrypt the first HTTPS request for authentication purposes.

** When cookie-based authentication is used, the Web Proxy cannot authenticate the user for HTTPS, native FTP, and FTP over HTTP transactions. Due to this limitation, all HTTPS, native FTP, and FTP over HTTP requests bypass authentication, so authentication is not requested at all.

*** No surrogate is used in this case even though cookie-based surrogate is configured.

Related Topics

- Identification Profiles and Authentication

Tracking Re-Authenticated Users

With re-authentication, if a more privileged user authenticates and is authorized, the Web Proxy caches this user identity for different amounts of time depending on the authentication surrogates configured:

- **Session cookie.** The privileged user identity is used until the browser is closed or the session times out.
- **Persistent cookie.** The privileged user identity is used until the surrogate times out.
- **IP address.** The privileged user identity is used until the surrogate times out.
- **No surrogate.** By default, the Web Proxy requests authentication for every new connection, but when re-authentication is enabled, the Web Proxy requests authentication for every new request, so there is an increased load on the authentication server when using NTLMSSP. The increase in authentication activity may not be apparent to a user, however, because most browsers will cache the privileged user credentials and authenticate without prompting until the browser is closed. Also, when the Web Proxy is deployed in transparent mode, and the "Apply same surrogate settings to explicit forward requests" option is not enabled, no authentication surrogates are used for explicit forward requests and increased load will occur with re-authentication.



Note If the Secure Web Appliance uses cookies for authentication surrogates, Cisco recommends enabling credential encryption.

Credentials

Authentication credentials are obtained from users by either prompting them to enter their credentials through their browsers, or another client application, or by obtaining the credentials transparently from another source.

- Tracking Credentials for Reuse During a Session, on page 42
- Authentication and Authorization Failures, on page 42
- Credential Format, on page 42
- Credential Encryption for Basic Authentication, on page 43

Tracking Credentials for Reuse During a Session

Using authentication surrogates, after a user authenticates once during a session, you can track credentials for reuse throughout that session rather than having the user authenticate for each new request. Authentication surrogates may be based on the IP address of the user's workstation or on a cookie that is assigned to the session.

For Internet Explorer, be sure the Redirect Hostname is the short host name (containing no dots) or the NetBIOS name rather than a fully qualified domain. Alternatively, you can add the appliance host name to Internet Explorer's Local intranet zone (Tools > Internet options > Security tab); however, this will be required on every client. For more information about this, see How do I properly set up NTLM with SSO (credentials sent transparently)?

With Firefox and other non-Microsoft browsers, the parameters **network.negotiate-auth.delegation-uris**, **network.negotiate-auth.trusted-uris** and **network.automatic-ntlm-auth.trusted-uris** must be set to the transparent-mode Redirect Hostname. You also can refer to Firefox is not sending authentication credentials transparently (SSO) . This article provides general information about changing Firefox parameters.

For information about the Redirect Hostname, see Configuring Global Authentication Settings, on page 26, or the CLI command `sethostname`.

Authentication and Authorization Failures

If authentication fails for accepted reasons, such as incompatible client applications, you can grant guest access.

If authentication succeeds but authorization fails, it is possible to allow re-authentication using a different set of credentials that may be authorized to access the requested resource.

Related Topics

- Granting Guest Access After Failed Authentication, on page 38
- Allowing Re-Authentication with Different Credentials, on page 40

Credential Format

Authentication Scheme	Credential Format
NTLMSSP	**`MyDomain\jsmith`**
Basic	**`jsmith`** **`MyDomain\jsmith`** **Note** If the user does not enter the Windows domain, the Web Proxy prepends the default Windows domain.

Credential Encryption for Basic Authentication

About Credential Encryption for Basic Authentication

Enable credential encryption to transmit credentials over HTTPS in encrypted form. This increases security of the basic authentication process.

The Secure Web Appliance uses its own certificate and private key by default to create an HTTPS connection with the client for the purposes of secure authentication. Most browsers will warn users, however, that this certificate is not valid. To prevent users from seeing the invalid certificate message, you can upload a valid certificate and key pair that your organization uses.

Configuring Credential Encryption

Before you begin

- Configure the appliance to use IP surrogates.
- (Optional) Obtain a certificate and unencrypted private key. The certificate and key configured here are also used by Access Control.

Procedure

Step 1 Choose **Network > Authentication**.

Step 2 Click **Edit Global Settings**.

Step 3 Check the **Use Encrypted HTTPS Connection For Authentication** check box in the Credential Encryption field.

Step 4 (Optional) Edit the default port number (443) in the HTTPS Redirect Port field for client HTTP connections during authentication.

Step 5 (Optional) Upload a certificate and key:

a) Expand the Advanced section.
b) Click **Browse** in the Certificate field and find the certificate file you wish to upload.
c) Click **Browse** in the Key field and find the private key file you wish to upload.
d) Click **Upload Files**.

Step 6 Submit and commit your changes.

What to do next

Related Topics

- Certificate Management.

Troubleshooting Authentication

- LDAP User Fails Authentication due to NTLMSSP

- LDAP Authentication Fails due to LDAP Referral
- Basic Authentication Fails
- Users Erroneously Prompted for Credentials
- HTTPS and FTP over HTTP Requests Match only Access Policies that Do Not Require Authentication
- Cannot Access URLs that Do Not Support Authentication
- Client Requests Fail Upstream Proxy