

## How to Configure Single Sign-On (SSO)

<https://campus.barracuda.com/doc/4259872/>

Single Sign-On (SSO) is a mechanism where a single set of user credentials is used for authentication and authorization to access multiple applications across different web servers and platforms, without having to re-authenticate.

The SSO system acts as a web gate for all inbound web traffic. When a user initially attempts to access the website, the user is challenged to provide login credentials. If authentication succeeds, an SSO user session cookie is generated. The SSO user session cookie authenticates the user for a period of time. If the log in fails, the authentication request is rejected.

The SSO environment protects defined resources (websites and applications) by requiring the following steps before granting access:

- **Authentication:** Authentication verifies the identity of a user using login credentials.
- **Authorization:** Authorization applies permissions to determine if this user may access the requested resource.

The Barracuda Web Application Firewall supports both single-domain and multi-domain SSO.

### Single-Domain SSO

Single-domain SSO takes place within a single domain. For example, consider the domain `barracuda.com`, which hosts several restricted websites on several hosts. You can configure single sign-on for this domain, so that authenticated users can access all or a subset of the restricted resources by authenticating just once.

#### Set Up a Single-Domain Single Sign-On Environment

1. Go to the **ACCESS CONTROL > Authentication** page.
2. Identify the service for which you want to set up single sign-on and click **Edit** next to that service. The **Edit Authentication Policy** window opens.
3. In the **Edit Authentication Policy** section, set the **Status** to *On*.
4. Select the **Authentication Service** to be associated with the service.
5. Set the **Session-Cookie Domain** to the domain name of the services for which you are configuring single domain SSO. Example: `service1` and `service2` both have `.barracuda.com` as the session cookie domain.
6. Click **Save**.

In a single-domain SSO set up, ensure that you configure the same **Session Cookie Domain**

name for each service on the **ACCESS CONTROL > Authentication** page.

### Logging Out in a Single-Domain Single Sign-On Environment

When a user logs out of a domain, the Barracuda Web Application Firewall removes the user session cookie from the browser by expiring it, so the user is automatically logged out of other corresponding domains. For example, consider a user logged into `host1.bc.com`, `host2.bc.com` and `host3.bc.com` using `bc.com` as the cookie domain. If the user performs a logout in `host1.bc.com`, the user session cookie is removed from the browser, and the user is automatically logged out of `host2.bc.com` and `host3.bc.com`.

If the user does not access the SSO environment within the specified idle timeout, the user session becomes idle, and the user is challenged to provide login credentials to access the SSO environment again.

## Multi-Domain SSO

Multi-domain SSO enables a user authentication to be honored by hosts in two or more domains. For example, a set of URLs that reside within the domains `www.abc.com` and `www.xyz.com` can be set to single sign-on.

To achieve a multi-domain single sign-on, a master domain is required for authentication. The Barracuda Web Application Firewall multi-domain single sign-on environment can have one master domain and one or more slave domains. The master domain acts as a centralized authentication server that authenticates the users and transfers the SSO user session cookie to the slave domains.

In a multi-domain single sign-on environment, each domain is responsible for maintaining and enforcing its own idle timeout. This means the cookie value for different domains might be different. You must configure the master service and the slave services on the Barracuda Web Application Firewall on the **ACCESS CONTROL > Authentication** page.

### Multi-Domain Single Sign-On Configuration

For a multi-domain SSO environment, specify the master service and master service URL for the domains as explained below:

- **Master Service:** Specifies if the master service URL is handled by this service. When set to *Yes*, this service acts as the master domain to the subsequent domains. When set to *No*, this service acts as the slave domain that accepts the cookie from the master domain.

- **Master Service URL:** Specifies the URL that provides a cookie. In case of the master domain, specify only the URL path. For the slave domains, specify the protocol, host, master domain, and URL path.

- The master service URL should be a virtual URL (internal URL). For example, `/ncsso.process`, `/index.html`, etc. This URL is used to identify the master service URL in a multi-domain environment.
- A global ACL rule must be created for the specified master service URL on the **SECURITY POLICIES > Global ACL's** page with the following configuration settings:
  - The parameter **Action** must be set to *Allow*.
  - The configured master service URL should be specified in the **URL Match** field.

For example, consider `www.abc.com` as the master domain and `www.xyz.com` as the slave domain. If the master service URL for the master domain is `/ncsso.process`, then the master service URL for the slave domain is `http://www.abc.com/ncsso.process`.

### Multi-Domain Single Sign-On Functionality

If a user attempts to access the master domain first, the user is challenged to provide login credentials. On a successful login, the user gains access to the master domain and to the slave domains. But if a user attempts to visit the slave domain first, the Barracuda Web Application Firewall redirects the user to the master service URL for authentication, and challenges the user to provide login credentials. If successful, the user gains access and is redirected to the requested domain.

For example, consider `www.abc.com` as the master domain and `www.xyz.com` as the slave domain. If a user attempts to access the master domain first (`www.abc.com`), the user is challenged to provide login credentials. An SSO user session cookie is generated on a successful login. Now, the user gains access and can navigate to the slave domains using the generated session cookie without having to re-authenticate.

If a user attempts to access the slave domain first (`www.xyz.com`), the web application redirects the user to the master service URL for authentication. The user is challenged to provide login credentials. If successful, SSO user session cookies are generated for both domains (master and slave) and the user is allowed to access the slave domain.

### Set Up Multi-Domain Single Sign-On Environment

#### Configure Master Domain

1. Go to the **ACCESS CONTROL > Authentication** page.
2. Identify the service that you want to configure as a master domain and click **Edit** next to that service. The **Edit Authentication Policy** window opens.
3. In the **Edit Authentication Policy** section, set the **Status** to *On*.
4. Select the **Authentication Service** to be associated with the service.

5. Scroll down to the **Single Sign On** section and set **Master Service** to *Yes*. Note that this is the master service that provides the cookie for the subsequent slave domains.
6. Specify the URL path in the **Master Service URL** field. The master service URL path can be any URL you choose. Example: `/ncsso.process`, `/index.html`, etc. Note that a global ACL rule must be created for the specified master service URL on the **SECURITY POLICIES > Global ACL's** page. For more information, see [Multi-domain Single Sign-On Configuration](#).
7. Click **Save**.

#### Configure Slave Domain

1. Go to the **ACCESS CONTROL > Authentication** page.
2. Identify a service that you want to configure as a slave domain and click **Edit** next to that service. The **Edit Authentication Policy** window opens.
3. In the **Edit Authentication Policy** section, set the **Status** to *On*.
4. Select the **Authentication Service** from the drop-down list to associate with the service.
5. Scroll down to the **Single Sign On** section and set **Master Service** to *No*. Note that this is the slave service.
6. Specify the protocol, host, master domain and URL path in the **Master Service URL** field.
7. Click **Save**.

#### Chained Logout in a Multi-Domain Single Sign-On Session

If your logout is from the master domain, the Barracuda Web Application Firewall removes the master domain cookie from the browser by expiring it. If your logout is from the slave domain, the slave domain cookie is removed from the browser by expiring it, which redirects the user to the master domain, and informs the master domain to log out the user (remove the master cookie).

For example, consider the case where three domains `www.xyz.com`, `www.abc.com` and `www.def.com` are a part of a multi-domain SSO environment, with `www.xyz.com` as the master domain, and both `www.abc.com` and `www.def.com` as the slave domains. When a logout is from the master domain (`www.xyz.com`), the user session cookie is removed from the browser by expiring it, which automatically accomplishes the logout of the corresponding domains (`www.abc.com` and `www.def.com`).

When a logout is from a slave domain (e.g., `www.abc.com`), the slave domain expires its cookie, redirects to the master domain (`www.xyz.com`), and requests the master domain to expire its cookie and log out. Then `www.abc.com` redirects the user to log out from `www.def.com`. To achieve this, configure **Auth Logout Success URL** as `http://www.def.com/nclogin.submit?f_method=LOGOUT` for authentication of `www.abc.com`.

In this case, the **Auth Logout Success URL** in `www.abc.com` is `http://www.def.com/nclogin.submit?f_method=LOGOUT`. This assumes that 'nclogin.submit' is configured as the **login-processor-path** in `www.def.com`. Similarly for multiple slave domains, you need to configure the same settings in `www.def.com` for the corresponding next domain and so

on.

## Steps Involved in Chained Logout

1. User performs a logout on `www.abc.com`. `www.abc.com` expires its cookie, and redirects the user to `www.xyz.com`.
2. `www.xyz.com` expires its cookie and redirects the user back to `www.abc.com`
3. `www.abc.com` redirects the user to perform a logout on `www.def.com`
4. `www.def.com` expires its cookie and redirects the user to `www.xyz.com`
5. `www.xyz.com` simply redirects the user back to `www.def.com`, since its cookie has been expired in Step 2.
6. The SSO user session cookie of all the three domains has been removed from the browser.
7. This process can be extended for more slave domains by simply chaining the **logout-success-url** configuration in the authentication container.

If the user does not access the SSO environment within the specified idle timeout, the session becomes idle, and the user is challenged to provide login credentials to access the SSO environment again.

## Related Articles

[How to Configure SiteMinder Single Sign-On \(SSO\)](#)

© Barracuda Networks Inc., 2020 The information contained within this document is confidential and proprietary to Barracuda Networks Inc. No portion of this document may be copied, distributed, publicized or used for other than internal documentary purposes without the written consent of an official representative of Barracuda Networks Inc. All specifications are subject to change without notice. Barracuda Networks Inc. assumes no responsibility for any inaccuracies in this document. Barracuda Networks Inc. reserves the right to change, modify, transfer, or otherwise revise this publication without notice.