Drupal Shibboleth module

Drupal **shib_auth** module enables Shibboleth authentication for Drupal CMS.

> **Warning**
>
> This document is written for module version 3.3-x. Please consult the Change log for the revisions of this document for the previous releases.
>
> For documentation about the more recent 4.x version, please read DrupalShibbolethReadmeDev.

> **Warning**
>
> The following documentation assumes that:
>
> • You understand how Shibboleth works
> • You have successfully installed and configured Shibboleth SP on your host running Drupal.

Installation

1. Download module source for your Drupal version from the project page.
2. Uncompress archive to the `modules/` directory
3. Enable module at **`Administer -> Site building -> Modules`**

Compatibility

Module is being developed for Drupal 6.x. We have stopped backporting new features to the 5.x branch and Drupal 7 is not yet supported as long as it isn't the stable branch. If you want to contribute to development or porting, please contact **aai *AT* niif *DOT* hu**!

Both Shibboleth 1.3 and Shibboleth 2.x are supported, although some features might require Shibboleth 2.x.

Upgrading module

If you are upgrading from the same major version, you only need to overwrite the files within your `modules/shib_auth` directory, then run `update.php`.

Configuration

Configuring Shibboleth

You should be familiar with protecting resources with Shibboleth before using this module. (See Shibboleth Wiki) Please check that Shibboleth authentication is working for that location and all the necessary attributes are exported to the headers. You can enable DEBUG mode to dump the whole **$_SERVER** array. If you can see Shibboleth attributes there, you're fine.

In Shibboleth there are two modes for protecting resources:

- **Lazy Sessions**: session is only initiated if an application redirects user to the SessionInitiator URL. In this module, it is done by clicking the "*Login with Shibboleth*" link. Anonymous access is possible as well as using other authentication methods.
  - Detailed description of lazy sessions in Hungarian.
- **"Strict" Sessions** (normal sessions): users can only access Drupal content if they have a valid Shibboleth session. This case, no anonymous access can be granted (not even read-only) and you can not use any auxiliary authentication methods.

> **Warning**
>
> If you decide to use lazy sessions and you don't want your users to be able to log in with a password, you have to disable changing passwords.

Example Shibboleth configuration

Note: this example uses lazy sessions. Configuration for Shibboleth 1.3 is quite similar.

/etc/shibboleth/shibboleth2.xml snippet:

```
<RequestMapper type="Native">
  <RequestMap applicationId="default">
    <Host name="your.host.name">
```

```
      <Path name="location_of">
        <Path name="your_Drupal">
          <Path name="installation" authType="shibboleth" requireSession="false" />
        </Path>
      </Path>
    </Host>
  </RequestMap>
</RequestMapper>
```

Apache config file snippet (ie. `/etc/apache2/sites-enabled/your.host.name`, or you can even use `.htaccess` without the `<Location>` tags):

```
<Location /location_of/your_Drupal/installation>
  AuthType Shibboleth
  ShibRequireSession Off
  # the following single line is only valid for Shib2
  ShibUseHeaders On
  require shibboleth
</Location>
```

> **Figyelem**
>
> You **MUST** use `ShibUseHeaders On` if you use Shibboleth2 with **mod_rewrite**.
>
> mod_rewrite prefixes CGI environment variables with **REDIRECT_**, so you have to instruct Shibboleth2 to use headers instead.
>
> Shibboleth 1.3 always uses headers, therefore the `ShibUseHeaders` directive is invalid with Shibboleth 1.3.

DEBUG mode

If you enable DEBUG mode on the module configuration interface, you can dump the whole **$_SERVER** array. This shows you all the available attributes and helps you diagnosing possible Shibboleth attribute problems. * Keep in mind that some users might have a specific attribute while others don't.

Debug path prefix

Leave it empty, if you want to display debug information on every page. For example use `user/` for display DEBUG messages on paths `user/*`

Adding a prefix is useful, if you want to enable debugging on an online drupal installation without littering all of the pages with the debugging information. Can be set to a non-existent node as well, in this case, the information will be displayed over the built-in 404 page.

Setting Shibboleth parameters for the module

Handler settings

If you are using lazy sessions, you have to define the Shibboleth SessionInitiator to which the user should be directed when she clicks on "Login with Shibboleth". SessionInitiator URL is constituted of the following:

- protocol scheme (`http://` or `https://`)
- host name
- Shibboleth handler URL (usually: `/Shibboleth.sso`)
- 'location' part of the SessionInitiator definition

/etc/shibboleth/shibboleth2.xml snippet:

```
<Sessions lifetime="28800" timeout="3600" checkAddress="false"
          handlerURL="/Shibboleth.sso" handlerSSL="false"
          exportLocation="http://localhost/Shibboleth.sso/GetAssertion"
          idpHistory="false" idpHistoryDays"7">
  <SessionInitiator type="Chaining" Location="/Login" isDefault="true" id="Intranet"
                    relayState="cookie" entityID="https://idp.example.org/shibboleth">
    <SessionInitiator type="SAML2" defaultACSIndex="1" template="bindingTemplate.html"/>
    <SessionInitiator type="Shib1" defaultACSIndex="5"/>
  </SessionInitiator>
  <!-- other things -->
</Sessions>
```

For this example, you should have:

- **`/Shibboleth.sso`** for *Handler URL*
- **HTTPS** or **HTTP** for *scheme*, depending on whether you are using SSL or not
- **/Login** for *WAYF location*

Attribute settings

Specify here the **$_SERVER** headers to look up the user's username and e-mail address. Please check **DEBUG** mode to look for the available headers. If you can not find the desired attribute, then something is wrong with your IdP-SP attribute release flow.

Both fields can have the same value, if you wish.

Using custom e-mail address

- *Require and use only Shibboleth-provided e-mail address* (default): with this option set, Drupal e-mail address is rewritten with the Shibboleth-provided one on each login. This means that your users can only use the e-mail address which is provided by the IdP. **The IdP is required to send the e-mail address attribute otherwise the user gets a fatal error.**
- *Ask for missing e-mail address*: let the user modify her own e-mail address by editing her Drupal account. If the IdP provides an e-mail address, then that value will be the default, otherwise the user is asked to specify her e-mail address.

Logging out

Session expiry

Enable the option "*Destroy Drupal session when the Shibboleth session expires*", if you want to force logout the users without a valid Shibboleth session. (This only applies to lazy sessions, otherwise you are always having a Shibboleth session.)

> **Info**
>
> Keep in mind if you leave this option off:
>
> • If the Shibboleth session is lost, all the Shibboleth-derived attributes disappear, therefore the user loses the assigned Shibboleth roles
> • On the other hand, the roles assigned to the *Drupal account of the user* persist as long as the Drupal session is valid
> • Shibboleth session might get lost if you use a clustered SP without a central session cache

URL to redirect to after logout

Define an URL here, where you want the user to be navigated after logout. The URL can be absolute or relative to the server base url. The relative paths will be automatically extended with the site base URL.

SAML2 Logout

At the moment, Shibboleth2 SP supports SAML2 logout while the Shibboleth2 IdP does not. It has a consequence that (if you have a standard Shibboleth2 installation), you will get a Shibboleth error message on logout, like this:

```
Global Logout
Status of Global Logout: Identity provider does not support SAML 2 Single Logout protocol.
```

You can avoid this message by commenting out SAML2 global logout initiator from `/Logout` handler in `/etc/shibboleth/shibboleth2.xml`:

```
<!-- LogoutInitiators enable SP-initiated local or global/single logout of sessions. -->
<LogoutInitiator type="Chaining" Location="/Logout" relayState="cookie">
  <!-- The following line should be commented out to make Drupal logout work,
       as long as your IdPs do not support SAML2 logout -->
  <!--LogoutInitiator type="SAML2" template="bindingTemplate.html"/-->
  <LogoutInitiator type="Local"/>
</LogoutInitiator>
```

Automatic role assignment

It's possible to assign roles to users based on their Shibboleth attributes.

An assignment rule is made of three parameters:

- **$_SERVER** header name: name of the Shibboleth-derived attribute
- **Value regexp**: regexp applied to (all) the value(s) of the Shibboleth-derived attribute
- **Role(s)**: checklist of roles to be assigned for the matching users

All these rules are evaluated at module initiation time. That means that revoking/adding a Shibboleth attribute rule will take effect immediately on next page refresh. The same applies when the set of headers is happened to be changed.

Additional roles can be assigned statically to the user (as an individual) by the administrator as normally.

> **Figyelem**
>
> Dynamic roles are not visible on the role administration page and on the user page. These roles are evaluated dynamically and are not saved to the database.

Using module

Automatic user creation

Drupal CMS requires all users to be in its internal SQL database. If the module detects that no user exists in the database with the received Shibboleth user identifier, it creates a new (Drupal) user.

Disallowing password change

There is no way for the module to detect if a user has been deleted from Shibboleth. This simple fact has a number of consequences.

When a user is first logged in, a Drupal account is automatically created for her. Because Drupal requires a password, a random string is generated for password. Normally the user doesn't need it.

Now suppose that your user is about to leave your institution. If she is malicious enough, she can go to the password change form, reset her password to a known one, and even after she is deleted from the IdP, she still can log in to your precious resource with the (now known) password. (Note that it is only achievable with lazy sessions!).

Therefore, if your requirements are such that only Shibboleth-authenticated users can log in, **YOU MUST DISABLE PASSWORD CHANGE** for users.

Steps for disallowing your users to change their passwords:

1. Install Drupal User Protect module
2. At Administer -> User management -> User Protect -> Protected roles tab check **password** for the *authenticated user* role.
3. At Administer -> Permissions -> userprotect module: uncheck **change own password** for *authenticated user*
4. Log in with a normal account, go to "My account" -> Edit. You shouldn't see the possibility for changing password; except for the case when the user has user administrator rights.

Administrator / password login

If you are using lazy sessions, you can still login with password. If you disabled the username/password login block, append the following to your normal Drupal URL: `/?q=user`

Administering Drupal with strict sessions

If you use strict sessions, you can not log in with a password. It's quite tricky to circumvent it:

1. Enable Shibboleth protection
2. Login with your own user credentials, so that your Drupal user profile is created
3. Disable Shibboleth protection
4. Login as 'admin', grant your own user 'Administrator' rights.
5. Enable Shibboleth protection
6. Login with your own credentials, you should have 'Administrator' rights now.

Change log

Version 3.2 -> 3.3

Module update problem was fixed. From now on one should run update.php on updates.

Previous version

Version 3.1 -> 3.2

The module now works with caching, but requires disabling and re-enabling.

Previous version

Version 3.0 -> 3.1

If you need documentation for 3.0, please use the previous version of the documentation

---

Változat #1
cziernorbert hozta létre 2026-04-14 13:22:37 CEST
cziernorbert frissítette 2026-04-14 13:23:01 CEST