Authentication and Single Sign On

Contents

1. Introduction
2. Fronter Authentication
   2.1 Passwords in Fronter
   2.2 Secure Sockets Layer
   2.3 Fronter remote authentication
3. External authentication through remote LDAP
   3.1 Regular LDAP authentication
   3.2 Cached LDAP authentication
4. Single Sign On methods
   4.1 Single Sign On with Fronter as authority (Fronter SSO)
     4.1.1 WSDL
     4.1.2 Methods
     4.1.3 Key
     4.1.4 Summary
     4.1.5 Example of implementation
   4.2 Single Sign On with external authority
     4.2.1 Scenario
     4.2.2 WSDL
     4.2.3 Methods
     4.2.4 Key
     4.2.5 Summary
5. Authentication with an external IMAP server
6. Shibboleth2 (SAML) SSO

Version control
Authentication and Single Sign On

1. Introduction

This document is an overview of the authentication and Single Sign On routines available in the Fronter Software. The document is intended as guidance for Fronter's partners, clients and resellers.

The scope of this document is exclusively authentication and SSO. Please refer to UAS and Interoperability documents for further details on identity management and user provisioning.

2. Fronter Authentication

Authentication is based on 1) something you know, 2) something you have in your possession or 3) something you are. As most web based applications, Fronter currently supports the first of these three methods. This means that you know your password in order to login to Fronter.

The password is either stored in the Fronter database or in a remote service. It is an advantage to have the password stored in a remote management system, since this will more easily enable a single-sign-on between more systems.

2.1 Passwords in Fronter

If the users are created manually in Fronter, or imported users get their usernames and/or passwords provisioned by Fronter, the passwords are stored in the database and are encrypted with MD5. It is also possible to import encrypted password from other systems. When the user logs on, password and username are always sent from the browser with SSL encryption.

In Fronter, all administrator users must have a complex password.

The rules for a valid password are:

- The password must be at least 8 characters
Authentication and Single Sign On

- Must contain letters from 3 of the 4 families: Small letters, Capital letters, numbers and special characters
- Cannot contain the Scandinavian letters åöæø
- If the password only contains one capital letter, it cannot be the first letter
- If the password only contains one number, it cannot be last letter.

It is possible to force these rules on all users to ensure security. This is achieved through setting 932 – Password regime, set to “Advanced” in the installation’s global settings.

2.2 Secure Sockets Layer

SSL is a protocol used for encrypted data transfer. The protocol is situated between TCP/IP and application level protocols such as HTTP, LDAP or IMAP. The purpose of encrypted communication is to authenticate the server and/or the client, and is used to ensure that the data transmitted is kept confidential. In addition it ensures that the data is not altered during transmission, which may be the case of non-encrypted transfers.

SSL communication over HTTP is normally done through port no 443. Thus, firewalls and other security instalments must allow data transfers to and from Fronter on this port.

2.3 Fronter remote authentication

Fronter has the ability to authenticate users through an external resource. The supported methods of remote authentication are

1. Remote “same sign-on” authentication with external LDAP(S) servers (openldap, Active Directory, E-Directory etc.)
2. Single sign on based on Web Services
3. Remote authentication with an external IMAP server
4. SSO with Shibboleth federations
Authentication and Single Sign On

3. External authentication through remote LDAP

3.1 Regular LDAP authentication

To enable LDAP authentication for users created or imported in Fronter, two steps must be completed:

1. Configure Fronter to communicate with your LDAP-server(s)
2. Change user accounts so the usernames match in both systems, and passwords connect to the LDAP authentication module

To complete the first step, a test form is available at https://fronter.com/[installationname]/testdata/ldaptest.phtml

The results of a successful test will then need to be sent to Fronter for adding onto the installation’s configuration. Please refer to the additional LDAP whitepaper, and the Fronter support site, for additional guidance and troubleshooting advice.

To complete the second step, all users must be imported with the same username as in the external source. Additionally, the password has to be set to a code “ldapX:” where “X” denotes the LDAP server number in case there are several servers configured. For example, “ldap1:” would be used for all users that should be authenticated by LDAP server number 1. If authentication should happen with more than one system, use “ldap2:”, “ldap3:” etc.

With this authentication method, the user’s password is always validated against the remote LDAP service, and not stored in Fronter at all.

3.2 Cached LDAP authentication

The LDAP password can also be cached in the Fronter database. Then, the password is validated against the remote LDAP service only when it does not match, or if the user has not logged in previously. This will provide a fail-over in the sense that any unavailability in the remote LDAP service will not prevent the majority of users from logging in.

The setup is similar to the regular LDAP one, but the password code is “ldapmd5:”.

fronter.com
Authentication and Single Sign On

4. Single Sign On methods

Fronter supports two SSO mechanisms for integrating with external applications. First where Fronter is the main service and second where external applications is the main service that the user logs into first. Both SSO mechanisms are built with Web-Service and SOAP technology.

4.1 Single Sign On with Fronter as authority (Fronter SSO)

4.1.1 WSDL
The WebService and it's methods are described in a WSDL (Web Service Description Language) file located at
http://fronter.com/[installation]/libs/ssoservice.wsdl.php

4.1.2 Methods
The identification service provides a method called "identify" that takes one argument "key". The identification service returns an identifier consisting of a name and a value. The identifier can be one of these: "username" or "id". If the identifier is username the value returned will be the username of the user holding the supplied key.

The selection of username or id to be used as identifier must be set by Fronter in the config file of the Fronter installation in question.

4.1.3 Key
The key is supplied to the external application in a URL (see example below). This key is then used in the call to the "identify" method of the identification service. The returned values will then identify the user logging in. The key is destroyed after the reply is sent to maintain high security. The key is randomly generated and therefore will be extremely hard to guess.

4.1.4 Summary
After the user is authenticated in Fronter, we contact the external application through an URL containing a keyword “#FRONTERSSOKEY#”. The keyword will upon launch be substituted with a randomly generated unique key. This key is linked to the user authenticated in Fronter. The external application then contacts Fronter through a
Authentication and Single Sign On

Web Service with this key as an argument. Fronter will then compare this key against the authenticated user and reply with the username of the user. The external application will now know that the user is authenticated in Fronter and can safely log him into the system. The key will then be destroyed for maintaining high security.

4.1.5 Example of implementation

The SSO links in Fronter resource folders are created so that a remote site has the possibility to check if the user is logged into Fronter. The usage is commonly as follows:

1. A link of the form http://yourservice.com?ssokey=#FRONTERSSOKEY# is put available as a link in a Fronter room. Note this has to be with the New link functionality in a resource folder.
2. When a user clicks on that link the “#FRONTERSSOKEY#” parameter is replaced with a generated id stored in the Fronter db. Use an additional parameter if the customer Fronter URL or any other data is also needed.
3. With that id the 3rd party can query Fronter to get a username (or userid).
4. The result can be used to grant or customise access to the remote service.

The code snippet below illustrates how steps 3 to 4 can be implemented.

```php
$url_in = $_SERVER['QUERY_STRING'];
parse_str($url_in, $data);
// ssokey is the SSO parameter in the URL the user has clicked in Fronter
$key = $data[ssokey];
// Replace [installation] with your proper installation URL
$sc = new SoapClient("http://fronter.com/
[installation]/libs/ssoservice.wsdl.php");
$result = $sc->identify($key);
$user = $result->value;
// You can now use $user to grant access etc.
```
Authentication and Single Sign On

4.2 Single Sign On with external authority

4.2.1 Scenario

Customers already having a system to be used as the master service for authentication must provide the specified identification service for Fronter. Fronter must be allowed to contact this identification service over the Internet from the fronter.com domain. The customer’s system can be any system as long as it provides the identification service.

In general the procedure is as follows

1. User logs in to the external resource/application
2. External resource generates a key for this login and saves it
3. External resource calls Fronter with this key
4. This key is used by Fronter to the identification service
5. The identification service looks up the key and finds out who the key belongs to
6. The returned values will then identify who is logging in and Fronter can grant access

A required first step to single sign-on with Fronter is to run a compatibility script. This is done at

http://fronter.com/[installationname]/testdata/webservice_test.phtml

The customer must pass this test to make sure schemas are compatible with Fronter SSO.

4.2.2 WSDL

For Fronter to be able to use the identification service an XML file describing the service must be provided. An example of a file that describes such a service is located at

http://fronter.com/[installationname]/libs/ssoservice.wsdl.php

The definition of the service endpoint in this file must be changed to reflect where the actual service is.
Authentication and Single Sign On

4.2.3 Methods
The identification service must provide a method called "identify" that takes one argument "key". The identification service returns an identifier consisting of a name and a value. The identifier can be one of these: "username" or "id". If the identifier is username the value returned will be the username of the user holding the supplied key.

The selection of username or id to be used as identifier must be set by Fronter in the config file of the Fronter installation in question, together with the remote WSDL endpoint URL.

4.2.4 Key
The key must be supplied to Fronter in a link (e.g. index.phtml?identification_key=key). This key is then used in the call to the "identify" method of the identification service. The returned values will then identify the user logging in. It is important that the key only can be used once to maintain high security.

The token name (e.g. "identification_key") is a parameter also set in the Fronter installation's config file.

The key can be anything, but it is advised to use something that is not readable, like an MD5 string.

4.2.5 Summary
The users login to the external resource/application. When logging in, the external resource generates a key specific to that login and saves it. The external resource then calls the link to Fronter with the key. The key is used by Fronter in the call to the identification service. The identification service looks up the key and finds out who the key belongs to. The returned values will then identify who is logging in and Fronter can grant access.

The relevant parameters that Fronter needs for configuration are the following:

- Whether the identifier will be username or id
- What the WSDL endpoint URL is
- What token name should be used for key (e.g. "identification_key")
5. Authentication with an external IMAP server

Each user can have a special email account that is used for external authentication. When a user logs into Fronter, the password is checked against the encrypted password stored in the Fronter database. If there is no match, a new check is made against the IMAP server. If this is accepted, the password is now stored encrypted in the Fronter database. The next time the user logs in with the same password, Fronter does not need to contact the IMAP server.

This can be useful since many organisations have IMAP servers in place already, and often have site wide passwords that apply to users' email accounts as well.

To enable IMAP authentication, the users' password has to be set to a code "imapmd5:“, and the remote IMAP authentication server has to be added on the installation’s configuration.

6. Shibboleth2 (SAML) SSO

Fronter supports the Shibboleth2 implementation of SAML. Whereas the solutions described above are suitable for integrating a single application or LDAP authentication service to your Fronter installation, SAML based solutions are usually federations between multiple organisations. Fronter can act both as a Shibboleth Service Provider and an Identity Provider.

To setup this kind of solution, you will need to have either a working SP or IdP, and provide Fronter with required metadata. Please contact your local Fronter office for further details, and the availability in your region.
# Authentication and Single Sign On

## Version control

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Description</th>
<th>Responsible</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>31.05.2005</td>
<td>First version</td>
<td>Bård Hall</td>
</tr>
<tr>
<td>0.2</td>
<td>11.08.2005</td>
<td>Added info about webservice</td>
<td>Bård Hall</td>
</tr>
<tr>
<td>0.3</td>
<td>25.08.2005</td>
<td>Changed link for WSDL test</td>
<td>Bård Hall</td>
</tr>
<tr>
<td>0.4</td>
<td>17.06.2009</td>
<td>Changed link for WSDL test</td>
<td>Aleksander Pettersen</td>
</tr>
<tr>
<td>1.0</td>
<td>25.03.2012</td>
<td>Rebranded, reviewed and updated contents</td>
<td>Tapio Janasik</td>
</tr>
<tr>
<td>1.1</td>
<td>29.03.2012</td>
<td>Added sections on cached LDAP and Shibboleth</td>
<td>Tapio Janasik</td>
</tr>
<tr>
<td>1.2</td>
<td>24.08.2012</td>
<td>Added a code snippet and a few clarifications</td>
<td>Tapio Janasik</td>
</tr>
<tr>
<td>1.3</td>
<td>05.05.2015</td>
<td>Rebranded</td>
<td>Sarah Voit</td>
</tr>
</tbody>
</table>