PHP Integration

Contents

Introduction

This article provides example files for integrating Swivel with a PHP-based website. The solution provided here is just an example: you will need to modify it according to your needs to produce a working solution.

Prerequisites

This link provides an example PHP filter - see below for more details. This version has been updated to include PINpad support, and has had limited testing on PHP version 8.1.

This link is the previous version. It was tested using PHP 5.3, but does not include PINpad support, only TURing.

The following links are for older PHP solutions. Going forward, it is recommended that you use the first link. The following libraries will not be maintained further.

This link provides a PINsafe API library and a basic example login page.

This link points to the previous version of the library, which uses the HTTP library. As not all PHP implementations include this, and the functionality can be reproduced using just the cUrl library, this version will not be maintained in the future. It has been tested with PHP version 5.3 on a Linux server (Ubuntu), but to use it in PHP 5.3 under Windows, you will need to get hold of the appropriate libraries. Unfortunately, a previous link to the relevant library is no longer valid, so we cannot provide a suitable link at present.

PHP Requirements

PHP version 5.3 or later. Version 5.2 may also work, but has not been tested. Versions earlier than 5.2 are known not to be compatible. The latest version has only been tested in PHP version 8.1.

The latest solution uses the PHP modules DOM and cUrl. The earlier version also uses the HTTP module, as described above.

The following setting is required in php.ini:

```
allow_url_fopen = On
```

Example PHP Filter

The example code above shows how you might use the PHP library to protect a PHP-based website. It contains the following files:

- swivel_client.php - An enhanced version of the PHP API library.
- config.php - this file is used to read in the configuration settings.
- swivel_filter.php - this file should be included in every PHP page you want to protect.
- image.php - A TURing image proxy.
- pinpad.php - A PINpad image proxy.
- login.php - an example login page with no image support.
- loginTuring.php - an example login page with TURing support.
- loginPinpad.php - an example login page with PINpad support.
- loginPush.php - an example login page with Push support.
- logout.php - an example logout page.
- testPage.php - an example web page that uses the filter, demonstrating how it should be used.
- login.css - the stylesheet for the login page.
- swivel_push.js - JavaScript to support Push login.
- config.xml - the Swivel server settings file.

- Copy all of these files to a subdirectory on your web site, for example "/swivel".
- Edit config.xml and enter the correct settings for your Swivel server - see below for more details. You should also enter a random string for the Cookie secret - the longer and more random the better.
- Optional: move config.xml to a location outside your website. Edit swivel_client.php and change CONFIG_DOC to the full path of this file. If you leave config.xml on the website with the other files, it can be read by browsers, which might be considered a security risk.
- Edit swivel_filter.php and set $swivelPath to the relative URL of the swivel subdirectory ('/swivel' if you are using the location suggested above). Change the name of the login page as well, depending on which one you want to use.
- Edit every PHP page that you want to protect with Swivel authentication, and add the line

```
<?php require('../swivel/swivel_filter.php'); ?>
```

Note that the exact URL above depends on whereabouts in the website your file is. This example assumes your page is in a subdirectory off the root website. If it is several levels deep, you will need to prepend more '../' entries to get to the right directory.

Filter Configuration

The following is an example config.xml file. Each value will be described below:

```
<?xml version="1.0" encoding="UTF-8"?>
<config>
 <server>localhost</server>
 <port>8080</port>
 <context>pinsafe</context>
 <secret>secret</secret>
 <secure>1</secure>
 <passwords>0</passwords>
 <ignoreCertErrors>1</ignoreCertErrors>
 <cookieSecret>sdf9087d2345fv89hn!</cookieSecret>
 <cookieTimeout>30</cookieTimeout>
 <caInfoFile></caInfoFile>
</config>
```

- **server** - the (internal) host name or IP address of the Swivel appliance
- **port** - the port used to communicate with the Swivel application - typically it will be 8080
- **context** - the application context for the Swivel application - this will invariably be "pinsafe"
- **secure** - 1 if you are using https to communicate with the Swivel application (the default), or 0 if it is http.
- **ignoreCertErrors** - 1 if you are using https, and the certificate on the Swivel appliance is not fully trusted, or the certificate subject does not match the **server** parameter given above. Typically, this will be required if you are using https, unless you set a trust store as described below.
- **passwords** - 1 if you want to show a password field on the login page as well as the one-time code. Normally, you only want this if you are using Swivel passwords as well as PINs, or you are using the option to check repository password.
- **secret** - the shared secret for the Swivel Agent. See here for more information.
- **cookieSecret** - the seed used to encrypt the authentication cookie. Setting this to a large, random value will improve security.
- **cookieTimeout** - the length of idle time (in minutes) before the authentication cookie expires and the user has to re-authenticate. The default is 10.
- **caInfoFile** - this is the full path to a trust store containing the CA root certificates in PEM format. PHP does not provide such a store by default, so you will need to create one: see PHP documentation on curl for details. Alternatively, specify ignoreCertErrors as 1, in which case you do not need to provide this (and it will be ignored anyway).