

Table of Contents

1 API How to Guide	1
1.1 Principles.....	1
1.2 Structure.....	1
1.3 Commands.....	2
1.4 Examples.....	2
2 Java Client	3
2.1 Introduction.....	3
2.2 Using the Library.....	3
3 Joomla 1.6 Integration	4
4 Introduction	5
5 Prerequisites	6
6 Baseline	7
7 Architecture	8
8 Swivel Configuration	9
8.1 Configuring the RADIUS server.....	9
8.2 Enabling Session creation with username.....	9
8.3 Setting up Swivel Dual Channel Transports.....	9
9 Joomla RADIUS Configuration	10
9.1 Joomla RADIUS Authentication Plug-in Installation.....	10
9.2 Joomla RADIUS Authentication Plug-in Configuration.....	10
9.3 Joomla RADIUS Authentication Plug-in Testing.....	11
10 Joomla Login Page Customisation	12
10.1 Creating a PINsafe login module.....	12
10.2 Adding the Custom Script.....	12
10.3 Configuring the PINsafe login module.....	13
10.4 Menu Assignment.....	13
11 Testing	15
12 Additional Configuration Options	17
13 Troubleshooting	18
14 Known Issues and Limitations	19
15 Additional Information	20
16 Microsoft IIS version 6 Integration	21
16.1 Overview.....	21
16.2 Prerequisites.....	21
16.3 PINsafe Configuration.....	21
16.4 Configuring the IIS Server.....	22
16.5 Configure the ISAPI Filter.....	24
16.6 Configure the ISAPI filter (Version 1.0-1.1).....	29
16.7 Installing the Filter on Multiple Websites.....	30
16.8 Testing.....	30
16.9 Troubleshooting.....	31
17 Microsoft IIS version 7 ASP.NET Forms Integration	33
17.1 Introduction.....	33
17.2 Prerequisites.....	33
17.3 Baseline.....	33
17.4 Architecture.....	33
17.5 ASP.NET Sample Files.....	33
17.6 PINsafe Configuration.....	33
17.7 ASP.NET Configuration.....	34
17.8 Additional Configuration Options.....	35
17.9 Testing.....	35
17.10 Troubleshooting.....	35
17.11 Known Issues and Limitations.....	35
17.12 Additional Information.....	35
18 Microsoft IIS version 7 ASP.NET Integration	36
18.1 Introduction.....	36
18.2 Prerequisites.....	36
18.3 Architecture.....	36
18.4 PINsafe Configuration.....	36
18.5 Filter Installation.....	37
18.6 Filter Configuration.....	37
18.7 Additional Configuration Options.....	43
18.8 Testing.....	43
18.9 Troubleshooting.....	43
18.10 Known Issues and Limitations.....	43
18.11 Additional Information.....	43

Table of Contents

19 Microsoft IIS version 7 Integration.....	44
19.1 Overview.....	44
19.2 Prerequisites.....	44
19.3 IIS Filter Version History.....	44
19.4 Swivel Configuration.....	44
19.5 Configuring the IIS Server.....	46
19.6 Installing the Filter on Multiple Websites.....	54
19.7 Testing.....	55
19.8 Uninstalling the filter.....	56
19.9 Troubleshooting.....	56
20 Microsoft RD Web Access.....	58
21 Introduction.....	59
22 Prerequisites.....	60
23 Swivel Server Configuration.....	61
24 Installation.....	62
25 Configuration.....	63
25.1 Configuration Options.....	63
26 Changes to Existing Files.....	66
27 Troubleshooting.....	67
28 Uninstalling.....	68
29 Microsoft Sharepoint 2010 Integration.....	69
29.1 Overview.....	69
29.2 Downloading the Filter.....	69
29.3 Upgrading the Swivel SharePoint Filter.....	69
29.4 Uninstalling the Swivel SharePoint filter.....	69
29.5 SharePoint PINsafe FAQ.....	69
29.6 Troubleshooting.....	69
30 Microsoft Sharepoint 2013 Integration.....	71
30.1 Overview.....	71
30.2 Downloading the Filter.....	71
30.3 Upgrading the Swivel SharePoint Filter.....	71
30.4 Uninstalling the Swivel SharePoint filter.....	71
30.5 SharePoint PINsafe FAQ.....	71
30.6 Troubleshooting.....	71
31 Microsoft Sharepoint 2019 Integration.....	73
31.1 Overview.....	73
31.2 Downloading the Filter.....	73
31.3 Upgrading the Swivel SharePoint Filter.....	73
31.4 Uninstalling the Swivel SharePoint filter.....	73
31.5 SharePoint PINsafe FAQ.....	73
31.6 Troubleshooting.....	73
32 Microsoft Sharepoint Integration Methods.....	75
32.1 Overview.....	75
32.2 Integration Using TMG or ISA.....	75
32.3 Authenticating to Earlier Versions of SharePoint as a 2-Stage Process.....	75
33 Open ERP7 Integration.....	76
34 Overview.....	77
35 Prerequisites.....	78
36 Swivel Configuration.....	79
37 Configuring the OpenERP 7 server.....	81
37.1 Upload the custom files to OpenERP 7.....	81
37.2 Testing.....	81
38 Error Messages.....	83
38.1 On OpenERP stack trace.....	83
38.2 On Swivel Log.....	83
39 OpenERP Custom Integration.....	84
40 Overview.....	85
41 Prerequisites.....	86
42 Swivel Configuration.....	87

Table of Contents

43 OpenERP User Authentication flow Integration.....	89
43.1 The Standard OpenERP User Authentication flow.....	89
43.2 The Swivel Integrated OpenERP User Authentication flow.....	89
44 Configuring the OpenERP Files.....	91
44.1 openerp/addons/web/static/src/xml/base.xml.....	91
44.2 openerp/addons/web/static/src/js/chrome.js.....	91
44.3 openerp/addons/web/static/src/js/coresetup.js.....	94
44.4 openerp/addons/web/controllers/main.py.....	95
45 Python 3.x changes.....	98
45.1 openerp/addons/web/controllers/main.py.....	98
45.2 Testing.....	98
46 Error Messages.....	101
46.1 On OpenERP stack trace.....	101
46.2 On the Swivel Log.....	101
47 Oracle WebLogic.....	102
48 Overview.....	103
49 Prerequisites.....	104
50 Baseline.....	105
51 Architecture.....	106
52 Installation.....	107
52.1 Swivel Integration Configuration.....	107
52.2 Configuring the Swivel Authentication Portal.....	107
52.3 Create private keys and certificates.....	108
52.4 Generating IdP metadata.....	108
52.5 WebLogic Integration Configuration.....	109
52.6 Additional Installation Options.....	120
53 Verifying the Installation.....	121
54 Uninstalling the Swivel Integration.....	122
55 Troubleshooting.....	123
55.1 Enabling WebLogic debugging.....	123
55.2 Error Messages.....	123
56 Known Issues and Limitations.....	124
57 Additional Information.....	125
58 PHP Integration.....	126
58.1 Introduction.....	126
58.2 Prerequisites.....	126
58.3 Example PHP Filter.....	126
59 Swivel Combined Client.....	128
59.1 Overview.....	128
59.2 Prerequisites.....	128
59.3 Using the Client.....	128
60 Website Integration.....	130
61 Website Authentication.....	131

1 API How to Guide

The Agent-XML API is an XML based API used for integrating PINsafe with other applications.

There are 4 subsets of the API, that cover the following areas

- Authentication

Authentication, Change PIN, PIN Reset, Start Authentication Session, Request security Strings etc

- Admin functions

Add new users, set user details (eg mobile phone number), synchronise a repository, delete users

- Helpdesk functions

Unlock user, send user new credentials, set user policies etc

- Reporting functions

Retrieving lists of idle users etc

All these APIs follow the structure described in this article.

1.1 Principles

The APIs are based around XML documents for both request and reply.

The basic idea is to build a document containing details of operation(s) to be performed and the user(s) on which they are to be performed, submit it to PINsafe via HTTP and PINsafe will reply with a document detailing which operations succeeded and which, if any, failed. All operations via the API will be logged in the standard PINsafe log.

Only authorised PINsafe Agents will be able to submit requests. This is in line with Agents used for authorisation. The API is case sensitive, the correct case shown in the included examples.

Requests are sent to PINsafe via an HTTP POST using http or https depending on the configuration of the PINsafe server. For appliances the default will be https over port 8080.

The requests are posted to the pinsafe context followed by AgentXML for the authentication API and AdminXML for the Admin, Helpdesk and reporting API.

For example `http://<ip address>:8080/pinsafe/AgentXML` or `https://<ip address>:8080/pinsafe/AdminXML`

1.2 Structure

The structure of a request that all API requests are contained within a request element. This request element specifies the type of request and also includes the shared secret for agent authentication.

eg an authentication request would be contained within an SASRequest tag

```
<?xml version="1.0" ?>
<SASRequest secret="MyAdminAgent" version="3.4">
  .
  .
</SASRequest>
```

Whereas an admin API request would be contained within an AdminRequest element.

```
<?xml version="1.0" ?>
<AdminRequest secret="MyAdminAgent" version="3.4">
  .
  .
</AdminRequest>
```

...or else a HelpdeskRequest element.

```
<?xml version="1.0" ?>
<HelpdeskRequest secret="MyAdminAgent" version="3.4">
  .
  .
</HelpdeskRequest>
```

The syntax of AdminRequest and HelpdeskRequest are more or less identical. The major difference is that AdminRequest can make major changes to the database, such as adding and deleting users, whereas HelpdeskRequest is largely read-only, although it does support some changes to existing users, such as PIN change. Conversely, AdminRequest can only affect users in the repository with the same name as the Agent, so the Agent must have "Can Act as Repository" set to Yes. HelpdeskRequest can affect any users.

The responses from the PINsafe server are similarly contained within response tags eg

```
<?xml version="1.0" ?>
<SASResponse secret="MyAdminAgent" version="3.4">
  .
  .
```

```
</SASResponse>
```

and

```
<?xml version="1.0" ?>  
<AdminResponse secret="MyAdminAgent" version="3.4">
```

```
</AdminResponse>
```

Within these elements for elements that specify the particular request. For details of these elements refer to the article relating to that part of the API.

1.3 Commands

Authentication

<changePIN> : Change Users Credentials

<exists> : Does a user have an account.

<login> : Perform PINsafe authentication

<ping> : Ping PINsafe application

<reset> : Perform a PIN reset

<resetcode> : Request a PIN Reset code

<startsession> : Start an authentication session

<securitystrings> : Request a set of security strings

Helpdesk functions

<reset> : Send user new credentials

<strings> : send dual channel security string(s) to user

<update> : Change user policies or credentials

Admin functions

<create> : create a new user and optionally set user details.

<delete> : delete a user.

<read> : read the details of a given user.

<reset> : Send user new credentials

<sync> : Synchronise with a repository

<update> : Change user policies, credentials, transports etc

Reporting functions

<disabled> : Report on disabled users

<idle> : Report on idle users

<locked> : Report on locked users

1.4 Examples

We have a number of applications that use these APIs, available as Windows executable programs. These can be found on a separate page.

2 Java Client

2.1 Introduction

This document describes the Swivel client library for Java applications.

The current version of the client library can be downloaded from [here](#).

2.2 Using the Library

Full documentation of the library is ongoing. Meanwhile, here are some examples of using the library:

2.2.1 Requesting a TURING image

The following code snippet will return a random security string as a TURING image:

```
PINsafeRequest req = new PINsafeRequest("https://swivel:8443/proxy");
req.singleChannelImageByUsername("user");
if (req.send()) {
    byte[] imageBytes = req.getResponseBytes();
    // imageBytes returns the TURING image as a JPG,
    // or if animation is enabled, an animated GIF.
}
```

2.2.2 Requesting a Dual Channel Message

The following code snippet will request a security string to be sent to the user's designated device:

```
PINsafeRequest req = new PINsafeRequest("https://swivel:8443/proxy");
req.dualChannelMessageByUsername("user");
if (req.send()) {
    byte[] confirmBytes = req.getResponseBytes();
    // imageBytes returns the confirmation image as a JPG.
    // This is only required if you wish to display this to the user.
}
```

2.2.3 Authenticating a User

The following code snippet will authenticate a user given the username and one-time code. It is assumed that the user has no Swivel password. If Swivel passwords are used, specify the entered password as the second argument to the login method. The authentication URL must be to pinsafe on port 8080.

```
// The first argument to the constructor is the PINsafe URL.
// The second argument is the agent secret.
AgentXmlRequest req = new AgentXmlRequest("https://swivel:8080/pinsafe", "secret");
// setIgnoreSSLErrors(true) will ignore SSL certificate errors.
req.setIgnoreSSLErrors(true);
req.login(user, "", otc);
if (req.send()) {
    boolean authenticated = req.actionSucceeded();
}
```

3 Joomla 1.6 Integration

4 Introduction

This document describes steps to configure Joomla with PINsafe as the authentication server.

To use the Single Channel Image such as the [TURing](#) Image, the PINsafe server must be made accessible. The client requests the images from the PINsafe server, and is usually configured using Network Address Translation, often with a proxy server. The Swivel virtual or hardware appliance is configured with a proxy port to allow an additional layer of protection.

5 Prerequisites

Joomla 1.6

PINsafe 3.x

[Joomla RADIUS Authentication plugin available here](#) Registration required.

When using the TURing, Security String Index or Message Confirmed, the required images are requested by the client from the PINsafe server. This is usually carried out through a NAT to the PINsafe server.

The PINsafe Joomla integration script can be found here: [PINsafe Joomla Integration Script](#)

6 Baseline

Joomla 1.6

PINsafe 3.8

7 Architecture

Joomla makes authentication requests against the PINsafe server by RADIUS.

8 Swivel Configuration

8.1 Configuring the RADIUS server

On the Swivel Administration console configure the RADIUS Server and NAS, see [RADIUS Configuration](#)

8.2 Enabling Session creation with username

To allow the TURing image, PINpad and other single channel images, under Server/Single Channel set [Allow session request by username](#) to Yes.

8.3 Setting up Swivel Dual Channel Transports

See [Transport Configuration](#)

9 Joomla RADIUS Configuration

A RADIUS module is used for Joomla authentication to authenticate to the PINsafe RADIUS server.

9.1 Joomla RADIUS Authentication Plug-in Installation

To install, on the Joomla Administration console select Extensions, then Extension Manager, there are 3 options for installation, select the desired installation method to upload the plugin:

- Upload Package File
- Install from Directory
- Install from URL

9.2 Joomla RADIUS Authentication Plug-in Configuration

When installation is complete on the Joomla Administration console select Extensions, then Plug-in Manager, the RADIUS plug-in should be listed.

<input type="checkbox"/>	Plug-in Name
<input type="checkbox"/>	Authentication - Joomla
<input type="checkbox"/>	Authentication - Radius
<input type="checkbox"/>	Authentication - GMail
<input type="checkbox"/>	Authentication - LDAP

Click on the plugin, and set the following information:

Enabled Enables the plug-in

Access Which level of access the plugin-is applied to all

Ordering In which order authentication is to be made

Details

Authentication - Radius

Enabled

Access

Ordering

Plug-in Type authentication

Plug-in File radius

ID 701

Description

Handles Radius user authentication

RADIUS Server The PINsafe server hostname or IP address

RADIUS Port The PINsafe server RADIUS port, usually 1812

Shared Secret The shared secret also entered onto the PINsafe server

other settings can be left as default depending on the required configuration

▼ Basic Options	
Radius Server	<input type="text" value="192.168.22.1"/>
Radius Port	<input type="text" value="1812"/>
Shared Secret	<input type="text" value="••••••"/>
Autocreate E-mail	<input type="text"/>
Email = Username:	<input type="text" value="Enable"/> ▼
Timeout in sec.	<input type="text" value="0"/>
Timeout in usec.	<input type="text" value="0"/>

When complete save the settings, ensuring that the plugin is enabled.

9.3 Joomla RADIUS Authentication Plug-in Testing

Test the RADIUS module with a username and password or OTC. A RADIUS request should be seen on the PINsafe server. A valid OTC can be derived from the PINsafe Administration console for a user by selecting View Strings.

10 Joomla Login Page Customisation

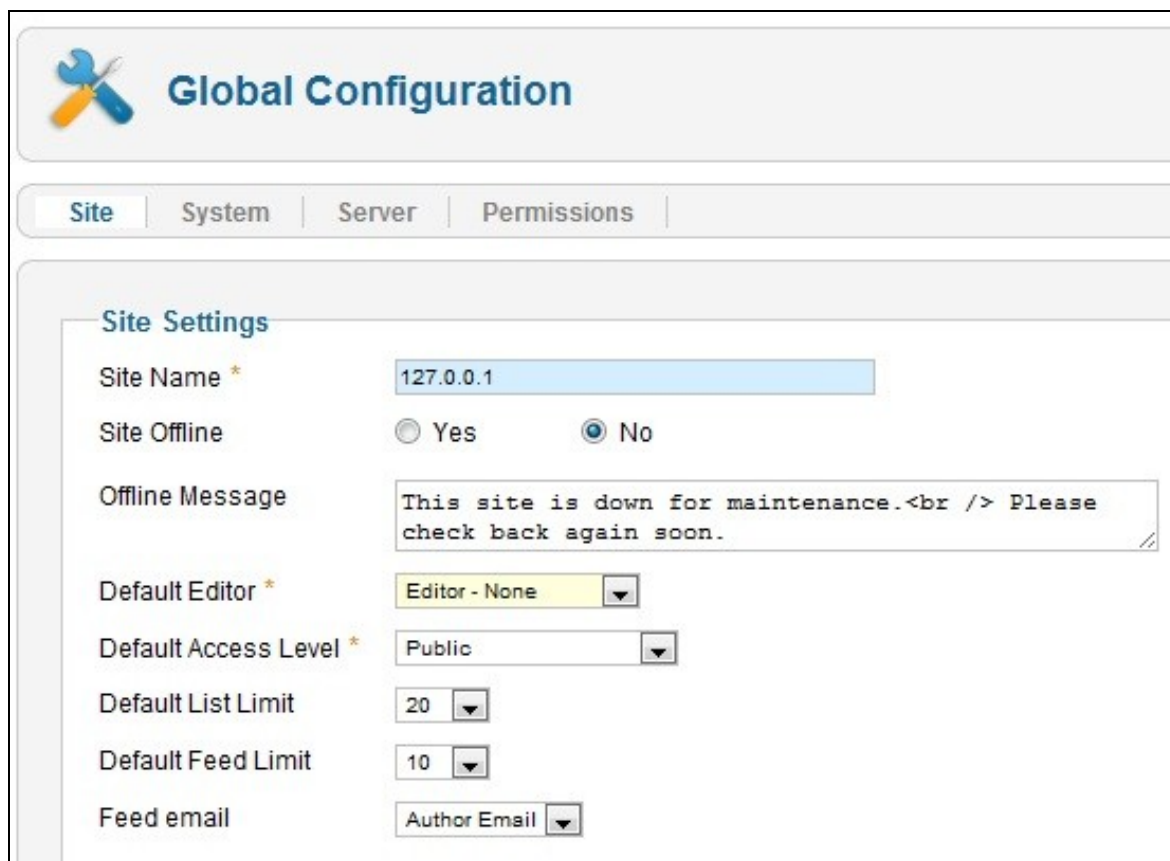
The Joomla login page can be modified in a number of ways, such as:

- Generation of Single Channel Images, such as TURing
- SMS Message request buttons
- Security String Index to show which security string can be used

10.1 Creating a PINsafe login module

A PINsafe login module can be downloaded and installed using [Joomla 1.6 PINsafe Module](#), or follow the instructions below to create a custom login module

In order to configure the PINsafe script, the WYSIWYG editor needs to be temporarily disabled. To disable/enable the editor, on the Joomla Administration console select Site, then Global Configuration, and set the Default Editor to Editor-None. If an error is received then the Administration Console permissions need to be correctly set. See [Cannot save Global Configuration changes](#)



The screenshot shows the Joomla Administration console's Global Configuration page, specifically the Site Settings tab. The page has a header with a wrench and screwdriver icon and the text "Global Configuration". Below the header are four tabs: "Site", "System", "Server", and "Permissions", with "Site" being the active tab. The main content area is titled "Site Settings" and contains several configuration options:

- Site Name ***: A text input field containing "127.0.0.1".
- Site Offline**: Radio buttons for "Yes" and "No", with "No" selected.
- Offline Message**: A text area containing the message "This site is down for maintenance.
 Please check back again soon."
- Default Editor ***: A dropdown menu set to "Editor - None".
- Default Access Level ***: A dropdown menu set to "Public".
- Default List Limit**: A dropdown menu set to "20".
- Default Feed Limit**: A dropdown menu set to "10".
- Feed email**: A dropdown menu set to "Author Email".

To create the new login module, on the Joomla Administration console select Extensions, then Module Manager. Click on New, and select a Module type of Custom HTML.

10.2 Adding the Custom Script

Under Custom output use enter the web page modification and script. The following lines need to be modified to reflect the environment

The following can be edited in the script to hide buttons that are not required:

TURing image button

```
<input type="button" value="TURing" onclick="showTuring();">
```

Show Security String Index button (To tell user which security string to use)

```
<input type="button" value="Index" onclick="showIndex();">
```

Message button to request a new security string to be sent to the user

```
<input type="button" value="Message" onclick="showMessage();">
```

The URL of the PINsafe server will also need to be modified to reflect the correct port and context.

For a virtual or hardware appliance installation:

pinsafeUrl = "https://turing.swivelsecure.com:8443/proxy/";

For a software only install see [Software Only Installation](#)

10.3 Configuring the PINsafe login module

Set the following details:

Title: PINsafe login. Descriptive Module Name

Show Title: Hide. Hides the Title in the login screen

Position: Position-7. This will vary according to the website design, and should be positioned close to the associated login module.

Status: published.

Access: Public. Select Access level appropriate

Ordering: 7. PINsafe Login. Where the PINsafe modification will appear in the login (this will depend on each site configuration)

Other settings can be left as default

Under Advanced Options, set Caching to None

The screenshot shows a configuration interface for the PINsafe Login module. The 'Details' section is expanded, showing the following settings:

- Title ***: PINsafe Login
- Show Title**: Show Hide
- Position ***: position-7 (with a 'Select position' button)
- Status**: Published (dropdown)
- Access**: Public (dropdown)
- Ordering**: 7. PINsafe Login (dropdown)
- Start Publishing**: 0000-00-00 00:00:00 (calendar icon)
- Finish Publishing**: 0000-00-00 00:00:00 (calendar icon)
- Language**: All (dropdown)
- Note**: (empty text box)
- ID**: 79 Custom HTML
- Site**: (empty text box)
- Module Description**: This Module allows you to create your own HTML Module using a WYSIWYG editor.

On the right side, the 'Advanced Options' section is partially visible, showing:

- Basic Options
- Advanced Options
 - Alternative Layout
 - Module Class Suffix
 - Caching
 - Cache Time

10.4 Menu Assignment

The module will need to be assigned, this will vary according to the site and page configuration. On the module, select Menu Assignment, then select the pages that are required. The below example uses the Module Assignment of 'Use only the pages selected' with Man Menu, Home selected.

Menu Assignment

Module Assignment

Only on the pages selected



Menu Selection:

Toggle Selection

About Joomla

Australian Parks

Fruit Shop

Main Menu

Top

User Menu

- Home
- Site Map
- - Articles
- - Weblinks
- - Contacts
- Login
- Sample Sites
- - Parks
- - Shop
- Site Administrator
- Example Pages

11 Testing

Connect to the Joomla website and verify that the correct images are shown

Login with TURing, String Index and Message Buttons

OTC Image

Login Form

User Name

Password

Remember Me

> Log in

[Forgot your password?](#)

[Forgot your username?](#)

[Create an account](#)

TURing Image Login



Login Form

User Name

Password

Remember Me

> Log in

Security String Index for SMS Message login

01

TURing Index Message

Login Form

User Name

Password

Remember Me

>Log in

SMS Message request Confirmed

TURing Index Message



Login Form

User Name

Password

Remember Me

>Log in

12 Additional Configuration Options

13 Troubleshooting

Check the PINsafe logs for Turing images and RADIUS requests.

14 Known Issues and Limitations

None

15 Additional Information

For assistance in the PINsafe installation and configuration please firstly contact your reseller and then email Swivel Secure support at support@swivelsecure.com

16 Microsoft IIS version 6 Integration

16.1 Overview

This document outlines the steps required to integrate the Internet Information Server (IIS) with PINsafe using dual or single channel authentication. The PINsafe install requires configuring an agent on the PINsafe server and setting up a shared secret with the IIS server to allow communication for authentication. An ISAPI filter installed on the IIS server allows access to protected resources through the PINsafe authentication.

NOTE: This document refers to the version of the filter numbered 1.1.0.1, and the configuration application with the same version number.

32 bit and 64 bit versions of the filter are available.

If Windows 2008 Server server is being used, or only ASP.Net applications are being protected an alternative authentication is available, see [Microsoft IIS version 7 ASP.NET Integration](#). However, this filter will still work in these situations if you prefer.

16.2 Prerequisites

Internet Information Server on Windows server 2000, 2003, 2008

PINsafe server

The appropriate PINsafe ISAPI filter software can be downloaded from here, depending on your operating system:

- [32-bit ISAPI Filter](#)
- [64-bit ISAPI Filter](#)

These links refer to the latest version of the filter: 1.3.8.

The previous version (1.2) is provided here:

- [32-bit ISAPI Filter](#)
- [64-bit ISAPI Filter](#)

16.3 PINsafe Configuration

On the PINsafe server configure the agent that is permitted to request authentication. On the PINsafe Administration Console select from the server menu Agents and enter the details of the IIS server IP address and a shared key, then click on apply.

Example:

```
Name : IIS server 1,  
Hostname/IP : 192.168.1.1,  
shared secret : secret
```

Agents:	Name:	<input type="text" value="local"/>	
	Hostname/IP:	<input type="text" value="127.0.0.1"/>	
	Shared secret:	<input type="password" value="....."/>	
	Group:	<input type="text" value="---ANY---"/>	
	Authentication Modes:	<input type="text" value="ALL"/>	<input type="button" value="Delete"/>
	Name:	<input type="text" value="IIS"/>	
	Hostname/IP:	<input type="text" value="192.168.1.1"/>	
	Shared secret:	<input type="password" value="....."/>	
	Group:	<input type="text" value="---ANY---"/>	
	Authentication Modes:	<input type="text" value="ALL"/>	<input type="button" value="Delete"/>

If Single Channel communication is to be used, select from the PINsafe Administration Console Single Channel, and set the Allow image request by username to Yes then click on apply.

Server>Single Channel

Please specify how single channel security strings are delivered.

Image file:	<input type="text" value="turing.xml"/>
Rotate letters:	<input type="text" value="No"/>
Allow session request by username:	<input type="text" value="Yes"/>
Only use one font per image:	<input type="text" value="Yes"/>
Jiggle characters within slot:	<input type="text" value="No"/>
Add blank trailer frame to animated images:	<input type="text" value="Yes"/>
Text Alpha Value:	<input type="text" value="80"/>
Number of complete display cycles per image:	<input type="text" value="10"/>
Inter-frame delay (1/100s):	<input type="text" value="40"/>
Image Rendering:	<input type="text" value="Static"/>
Multiple AUTHentications per String:	<input type="text" value="No"/>
Generate animated images:	<input type="text" value="No"/>
Random glyph order when animating:	<input type="text" value="No"/>
No. Characters Visible:	<input type="text" value="1"/>

16.4 Configuring the IIS Server

16.4.1 Install the PINsafellISFilter.exe

1. On the IIS server run the PINsafellISFilter.exe. The IIS filter may need to be run as an Administrator user (The filter needs to be installed by IIS running as Administrator user but it can run as a normal user).
2. Choose the Path to Install to - the default is as C:\Program Files\Swivel Secure\Swivel IIS Filter
3. Select Start Menu Folder
4. When details are correct click on Install
5. If the error ?Incorrect Command Line Parameters? is seen click on OK

NOTE: you will see that there are two installation options: "Filter" and "Configuration". Typically, you would install both on the web server, but the configuration program requires Microsoft.Net Framework 4.0 or higher installed. If your web server doesn't have this, and you prefer not to install it, then you can install the configuration program on a separate machine. You would then need to create the configuration file locally, and copy it to the web server.

16.4.2 Configure the ISAPI filter

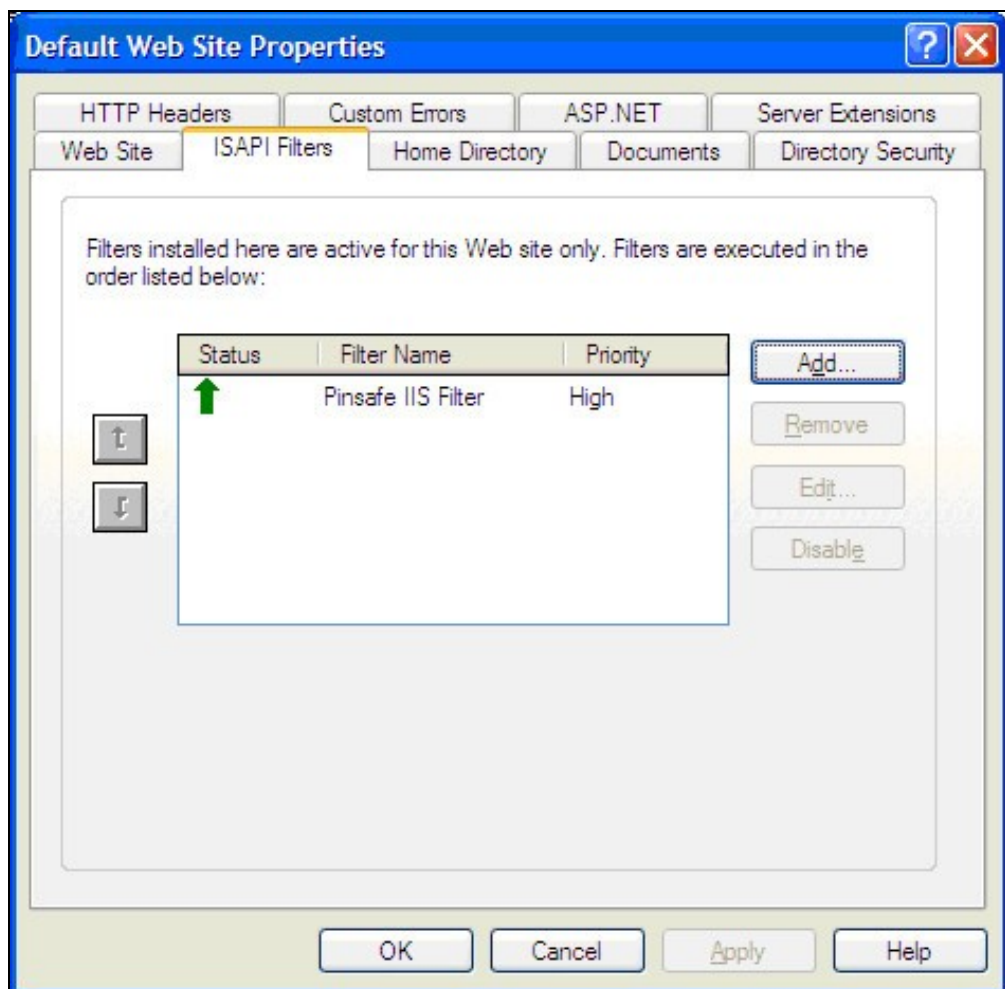
When the installation is completed, you will be presented with the configuration program. See below for details on using this.

16.4.3 Create a PINsafe virtual directory

1. On the Internet Information Services Manager right click on the website and select New, Virtual Directory
2. Create an Alias called PINsafe
3. Point the path to the PINsafe directory Web folder, by default C:\Program Files\Swivel Secure\Swivel IIS Filter\Web.
4. Set the permissions to Read and Run Scripts
5. Right-click on the newly-created virtual directory and choose Properties. On the Virtual Directory tab, click the Remove button next to Application name and then click OK.

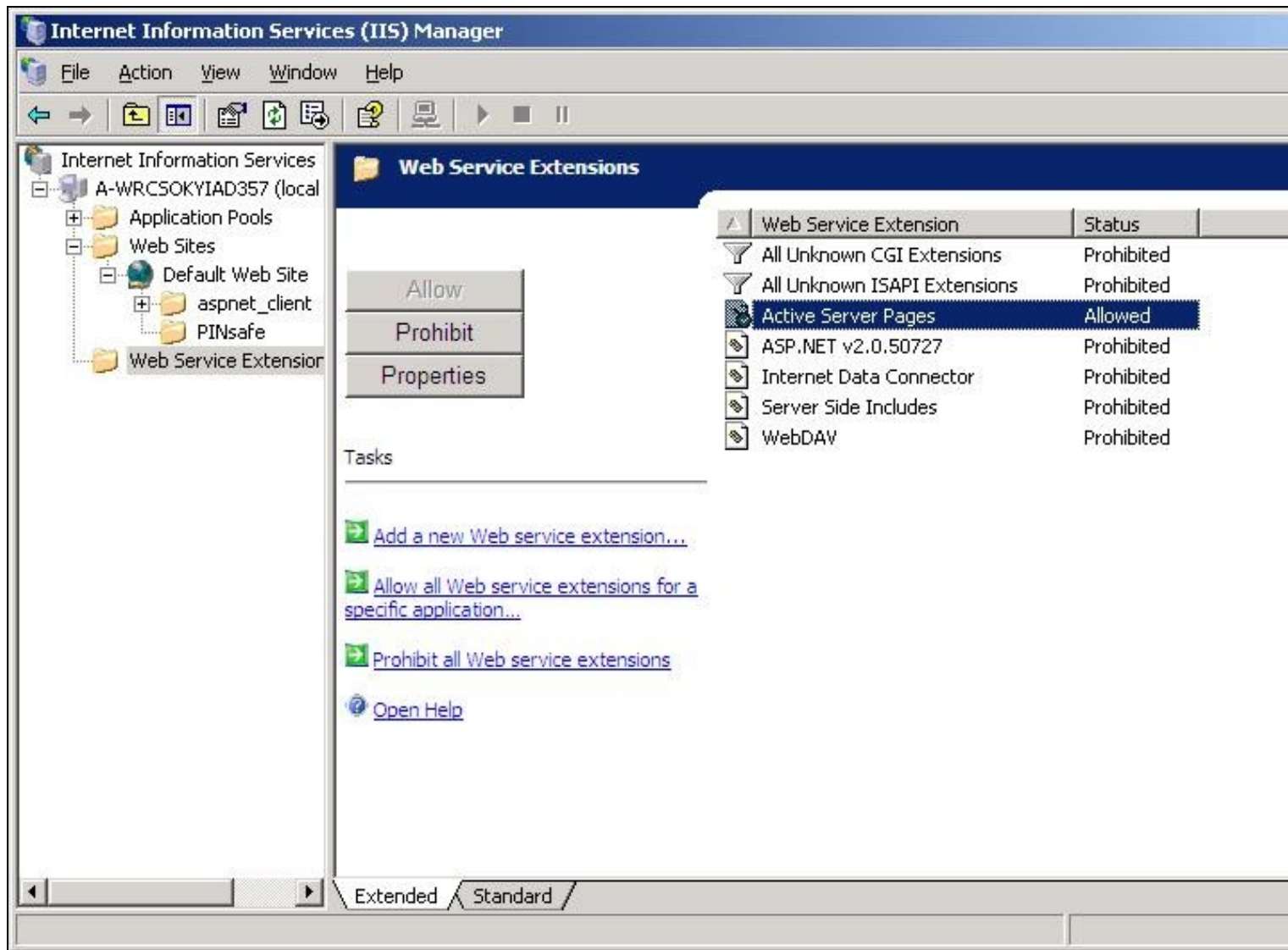
16.4.4 Install The IIS ISAPI filter

1. On the Internet Information Services Manager Select the Properties for the website
2. Select ISAPI filters
3. Select Add ISAPI filter
4. Select the Path to the PINsafe ISAPI filter. Note that the actual file you require will be PINsafellISFilter.dll, located in the installation folder.
5. Ensure PINsafe ISAPI filter is top filter then click on OK



From the Services application, usually found from the Control Panel Administrative Tools, restart the World Wide Web Publishing Services.

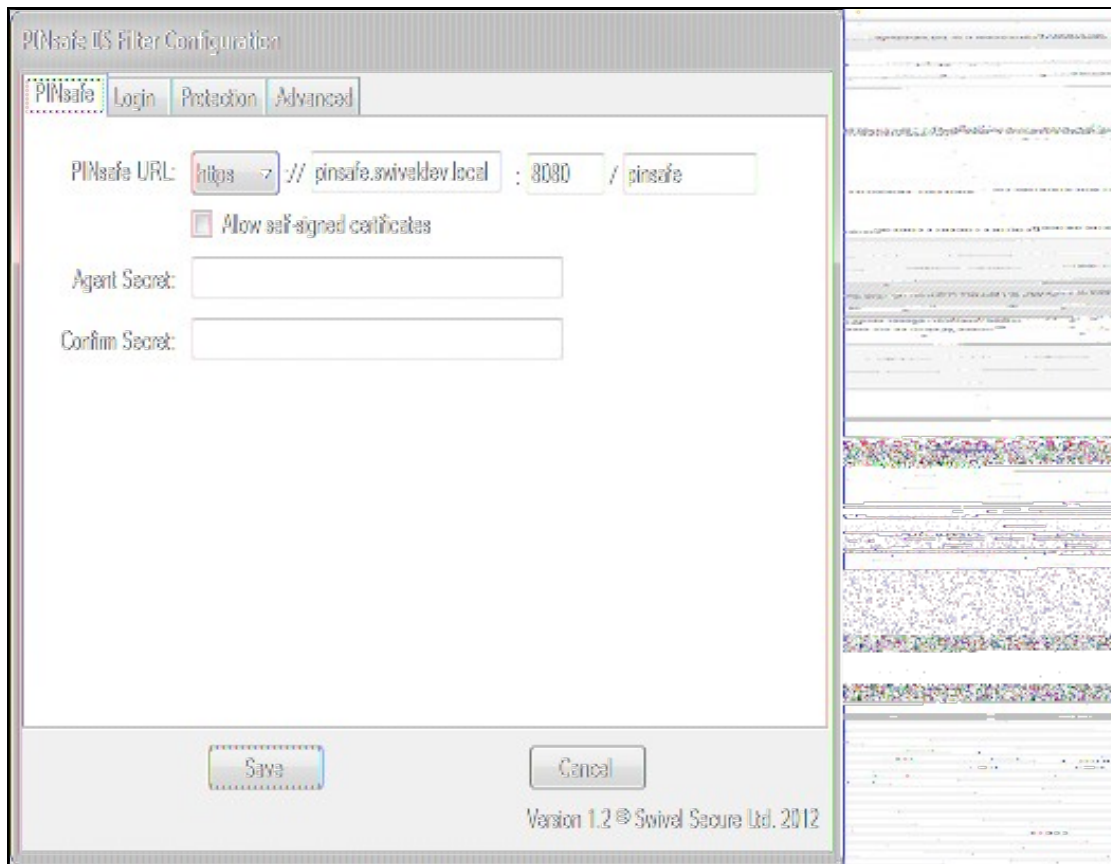
Allow Active Server Pages: to verify this, select the Internet Information Services Manager expand the required server then click on Web Service Extension.



16.5 Configure the ISAPI Filter

This documentation refers to version 1.2 of the configuration program. If you are still using an older version, see the next section for a description of the configuration program.

16.5.1 PINsafe Server Settings



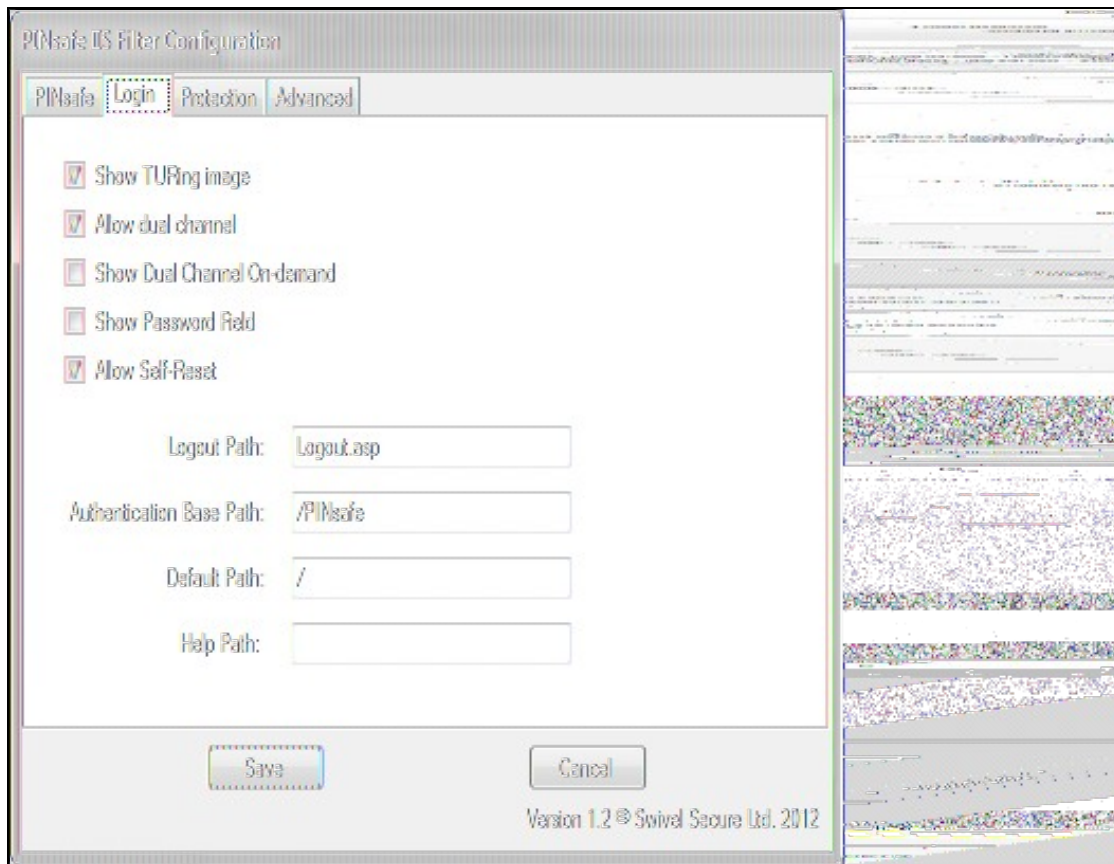
This page defines the connection to the PINsafe server.

In the first line, enter the URL for the PINsafe server. As you will see, it is entered in several parts: http/https, the server host name or IP address, port number and context.

The check box on the second line indicates whether self-signed SSL certificates are allowed for https. This actually ignores all SSL certificate errors, including incorrect host name and expired certificates. You should only use this option if the connection is internal only, and you are confident that the PINsafe server settings are correct.

The final option on this page is the shared Agent secret. This should be the same as the secret entered for the Agent entry on the PINsafe configuration. It is not normally displayed, and you should only enter a value if you wish to change it: a blank entry will result in no change. You need to enter the same value twice to ensure it is entered correctly.

16.5.2 Login Page Settings



This page defines how the login page is displayed, and what happens on login.

The first 5 checkboxes enable or disable features on the page:

Show **TURING** image: displays a button to show a TURING image.

Allow dual channel: has no obvious effect - dual channel authentication is always allowed if PINsafe policy permits it.

Show Dual Channel On-demand: displays a button to request an on-demand security string.

Show Password Field: requests a PINsafe password as well as the one-time code. This will also enable repository (e.g. AD) password if the Agent has "Check Repository Password" enabled.

Allow Self-Reset: shows a link on the page to the self-reset page, in case the user has forgotten their one-time code.

The four paths are:

Logout Path: if the filter detects this path, the PINsafe authentication cookie is removed, so the user must log in again.

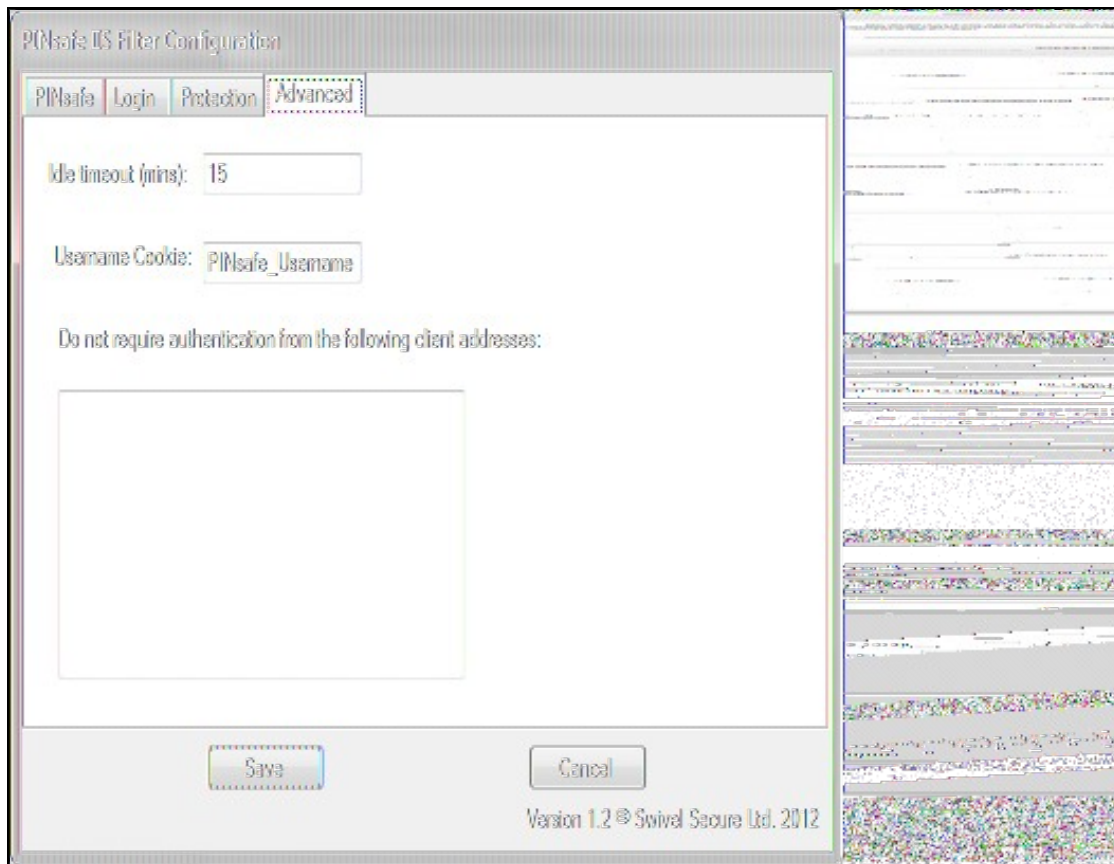
Authentication Base Path: the virtual path containing the PINsafe authentication pages.

Default Path: if a user navigates directly to the PINsafe login page, rather than being redirected by the filter, this is the path the user will be redirected to on successful authentication.

Help Path: if present, a link will be displayed to this path if the user requires help. This must be provided by the customer: Swivel does not provide any help pages.

16.5.3 Advanced Settings

Let us take the last tab out of order, as the Protection tab is the most complicated one:



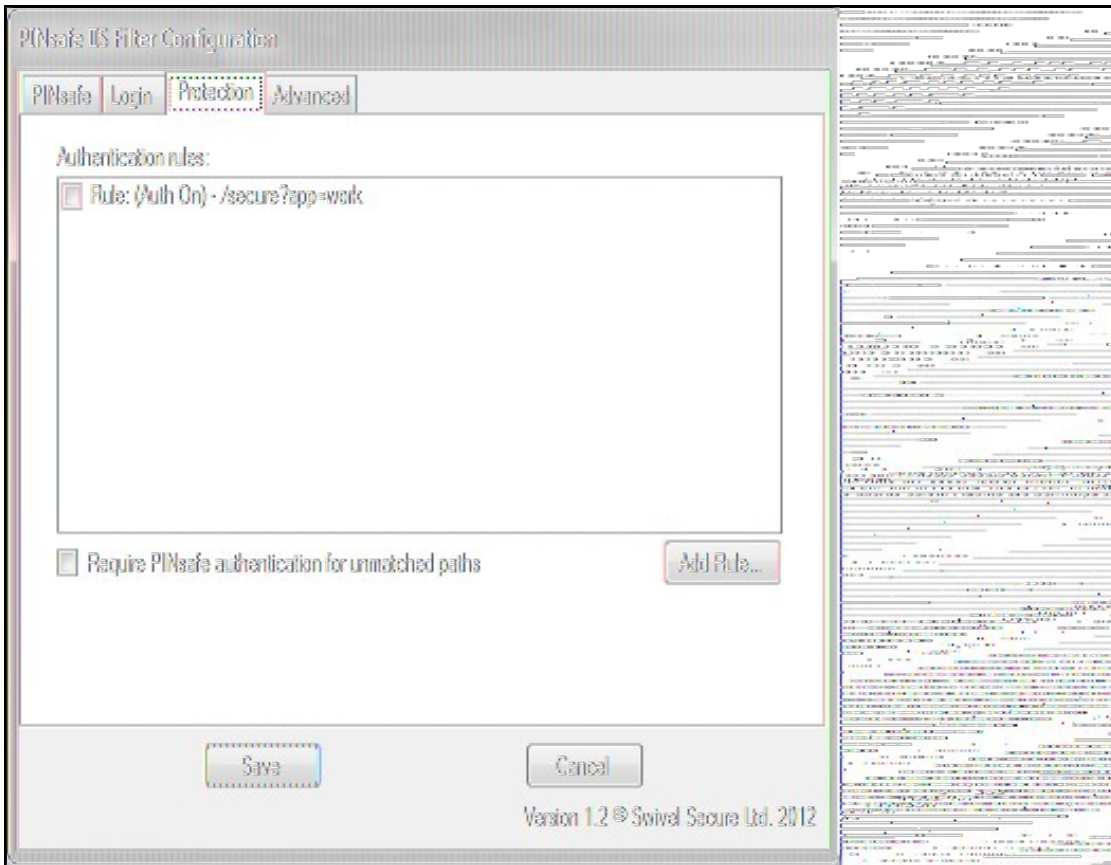
Idle timeout is the time (in minutes) that the user can leave a page open without refreshing it or navigating to another page: in other words, the lifetime of the authentication cookie. However, if the user requests a new page (or refreshes the current one) within that time, the cookie expiration time is updated.

Username cookie, if entered, specifies the name of a cookie that will contain the name of the authenticated user. Other applications can make use of this cookie if they are written to read it.

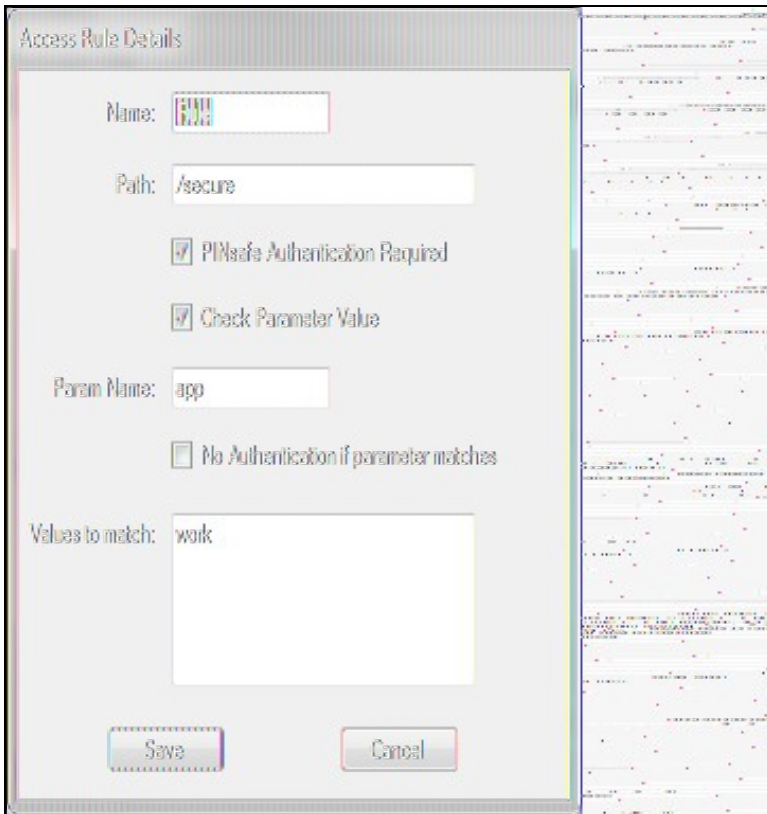
The final option on this page allows you to specify a list of source addresses that are not required to authenticate to PINsafe. Typically, these will be internal addresses.

16.5.4 Protection Settings

This tab replaces the Included and Excluded paths of the older filter:



In order to define which paths PINsafe protects, you need to define rules. The main part of this tab summarises the current list of rules. To add a new rule, click "Add Rule...", and you will see the following page.



The rule name is just a means of identifying the rule: it doesn't affect how the rule works.

The path is the URL that must match the URL entered for the rule to apply. The path must start at the slash immediately after the host name (and port if given). The match is case-insensitive, and the entire entered URL does not have to match the path: it just has to match as far as the path is specified. So, for example, if the path is "/secure", it will match "/secure/default.aspx", or even "/securepage", but not "/somewhere/secure".

The next checkbox indicates what happens if the path is matched. If it is checked, PINsafe authentication is required, and if no PINsafe cookie is found, the user is redirected to the login page. If this box is unchecked, the user is permitted to continue without authenticating, and no further rules are tested.

The remainder of the rule allows you to restrict PINsafe authentication according to the value of a particular parameter in the query string. Check the "Check Parameter Value" checkbox to enable this option.

Param Name is the name of the parameter that must be matched. Values to match allows you to specify a list of values that are accepted. The parameter must match one of these values.

The final checkbox defines how PINsafe authentication is affected depending on the value of this parameter. Normally, PINsafe authentication is applied if any of the values match. Checking this box reverses the logic, so PINsafe authentication is applied only if the parameter DOESN'T match any of these values.

Note that the parameter value only affects whether or not PINsafe authentication is applied, not whether or not the rule matches. Rule matching is done by path only.

Note also that parameter matching only applies to HTTP GET requests, i.e. when the query string is part of the URL. It cannot handle POST requests, when the parameters are in the body of the request.

So, using the example rule above: if the URL entered is "/secure/default.aspx?app=work", then PINsafe authentication is required. If the path is "/secure/default.aspx?app=play", or "/secure/default.aspx" (i.e. no parameter), then PINsafe authentication is NOT required.

NOTE: all comparisons, of path, parameter name and parameter value are case-insensitive.

The filter works by checking each rule in the order given. The first rule that matches determines whether or not PINsafe authentication is required for that URL.

You can change the order of the rules by right-clicking on the list. There are options to move sets of rules to the top or bottom, to move individual rules up or down the list, or to delete rules. You also use this menu to modify an existing rule. The dialog displayed is the same as above.

Finally, you can specify what happens if the entered URL doesn't match any rules: by default, no PINsafe authentication is required. If you check the final checkbox, PINsafe authentication will be required for all URLs that don't match any explicit rules.

16.5.5 Special Consideration for Windows Server 2003 / Windows XP

The settings are saved to the Windows common data folder. In Windows Server 2008 / Windows 7 and later, this is usually **C:\ProgramData**. In Windows Server 2003 and Windows XP or earlier, it is **C:\Documents and Settings\All Users\Application Data**.

The configuration program, and the filter itself, automatically select the correct folder. However, the web page **settings.asp** has the path hard-coded. If you are using Windows Server 2003 or earlier, or if you have changed the common data folder for some reason, you need to edit **settings.asp** to set the correct folder for **config.xml**. Edit the file **C:\Program Files\Swivel Secure\Swivel IIS Filter\Web\settings.asp** and look for the following line:

```
configDoc.load("C:\ProgramData\Swivel Secure\IIS Filter\config.xml")
```

Change the file path to the correct path for your environment.

16.5.6 Reading and Saving Configurations Elsewhere

The File menu on the configuration program allows you to save a copy of the configuration elsewhere, or to read a configuration file from elsewhere. This is useful if you are configuring the filter from a different machine, or if you have multiple configurations.

Additionally, you may find that you are unable to save the configuration to the default location (C:\ProgramData\Swivel Secure\IIS Filter\). You may find that the program appears to save it, but when you check, it has not been saved there. In this case, save a copy of the configuration file (**config.xml**) to a different location, and then copy it to the correct location.

You will also need to do this if you have installed the configuration program on a separate computer.

16.6 Configure the ISAPI filter (Version 1.0-1.1)

This documentation applies to the older version of the filter.

1. Select Start Programs/PINsafe IIS Filter/Filter Configuration. This will look for the location of **config.xml**, this will be created when first used and this must be located in **web/bin**.

2. You will be asked to select the location of the configuration file. It is important that you select the right location. It should be in the same folder as the ISAPI filter. Initially, this file will not exist, but will be created as a result of running this configuration application.

16.6.1 PINsafellISFilter Options

PINsafeServer: The PINsafe Server tab contains settings which define the PINsafe server which will be used to authenticate users.

Hostname/IP: The name or IP address of the PINsafe server.

Port: The port number used by the PINsafe server (normally 8080).

Context: The context (i.e. web application name) of the PINsafe instance on that server

Secret: The common secret used to communicate with the PINsafe server. This value must be the same as the secret defined for the PINsafe agent configured earlier.

SSL enabled: Tick this box to require SSL (HTTPS) communication with the PINsafe server.

Permit self-signed certificates: Tick this box to allow SSL certificates to be self-signed. This also ignores other certificate errors, such as site names not matching.

Authentication: The Authentication tab contains the following settings:

Idle time (s): The time (in seconds) for which the authentication cookie will be valid if the web page is not used.

Username header: The name of a cookie which will pass the username of the authenticated PINsafe user. If this value is blank, no cookie will be provided.

Single: Indicates that single channel security strings (i.e. TURing image) are permitted.

Dual: Indicates that dual channel security strings (i.e. via e-mail, SMS) are permitted.

On-demand dual: Indicates that the login page should display a button to request dual-channel security strings.

Display password fields: Indicates that the login page should show a field for PINsafe password as well as OTC.

Permit self-reset: Indicates that the user self-reset page should be enabled.

Exclusions and Inclusions: Use the inclusion and exclusion tabs to enter which paths should be protected by PINsafe:

Included paths: This is a list of paths within the current website which require PINsafe authentication. If this list is empty, the entire website will be protected except as indicated by the Exclusions tab. Paths should be one per line.

Excluded paths: This is a list of paths within the current website which should be exempt from PINsafe authentication. This is only relevant if the included paths list is empty, in which case all paths not on this list will be protected by PINsafe.

Excluded addresses: This is a list of IP addresses which are exempt from PINsafe authentication. All requests from these addresses are passed through without authentication.

Misc: On the Misc tab, edit any custom paths as follows:

Default path: This is the path to which authenticated requests are directed if the login page is targeted directly. If a user tries to access a protected page, she is redirected to the login page, and after authentication, back to the page she was trying to access. If the user requests the login page directly, she will be redirected to this location after authentication.

Logout path: Requesting this path will result in the user being logged out. Subsequent requests will require re-authentication, if relevant. If this path is empty, users can only be logged out by closing the browser, or if the authentication times out. Note: The logout path must be an included file location.

Virtual web path: This is the path to the PINsafe authentication pages. See the next section for details on setting this up. You should normally set this to be `?/pinsafe?`, unless you have a particular reason not to.

Help URL: The URL for PINsafe IIS filter help. The filter does not come with help pages as standard, so this should only be filled in if help pages have been provided by the reseller or end user.

16.7 Installing the Filter on Multiple Websites

Most of the instructions above assume that you are installing the IIS filter on the single default website of a web server. However, if you want to use the filter on multiple websites on the same server, you need to carry out a few extra steps. Firstly, if all the settings on all the websites are exactly the same, you can configure it once, and then carry out the same steps to activate the filter on each website. In this case, simply save the settings to all the relevant locations.

If, however, you need different settings for each website, you will need to do the following for all except the first website:

1. Make a copy of the entire Web folder, including the bin sub-folder. You can copy it as a sub-folder of PINsafe IIS filter, with a different name, or you can put it somewhere completely different. It is important, however, that you keep the structure the same ? all the .asp files must remain intact, and the filter DLL must be in a sub-folder called `?bin?`. You do not, however, have to copy the FilterConfig.exe file (but it does no harm if you do).
2. When you come to configure each filter, navigate to the bin sub-folder of the copy you have just made. Otherwise, configuration is exactly as before.
3. When selecting the IIS filter to install, and also when defining the virtual directory for PINsafe web pages, you should use the copy you have just created, rather than the original. This is important, as both the filter and the web pages look for the configuration file relative to their own location.

16.8 Testing

Browse to a web page that has been configured for protection. This should display a PINsafe login dialog:



The image shows a login dialog box with a light gray border. It contains three text input fields stacked vertically. The first is labeled 'Username:', the second 'Password:', and the third 'OTC:'. Below these fields are two buttons: 'Start Session' on the left and 'Login' on the right. The dialog box is centered on the page.

Enter the Username.

For dual channel, enter the One Time Code:

A screenshot of a web-based login form. It features three input fields: 'Username' with the text 'admin', 'Password' (empty), and 'OTC' with five black dots. Below the fields are two buttons: 'Start Session' and 'Login'.

Or click start session to enter a single channel OTC. The PINsafe log will record that a single channel session has started.

A screenshot of a web-based login form, identical to the one above. It features three input fields: 'Username' with the text 'admin', 'Password' (empty), and 'OTC' with five black dots. Below the fields are two buttons: 'Start Session' and 'Login'.

If authentication is successful it should redirect to the login page. If failed an error message will appear. The PINsafe log will record any successful log attempt for the agent.

A screenshot of a web-based login form with an error message. The form has three input fields: 'Username' with 'admin', 'Password' (empty), and 'OTC' (empty). Below the fields are 'Start Session' and 'Login' buttons. A large orange box with black text is overlaid on the bottom half of the form, containing the message: 'An error occurred, please check your credentials. If the error persists contact your PINsafe Administrator.'

16.9 Troubleshooting

Verify the correct filter version has been installed for the operating system i.e. 32 bit or 64 bit.

Check for error messages in the PINsafe log

Check the IIS log messages

Filter Install issues: The IIS service needs to run as an administrator in order to install the filter. Once the filter is installed, it will run as the normal user. Open the Services applet and locate World Wide Web Publishing. Select properties, then go to the Log On tab. Assuming you are logged in as an administrator, select Local System account. Then restart the service. Hopefully, this will start the filter. You can then switch back to the default account and restart again.

If you are not redirected to the PINsafe login page when trying to access a protected page, open IIS manager and check that the ISAPI filter is installed and has loaded properly (there should be a green arrow next to it, and the priority should be ?High?). If this is not the case, restart IIS, if you have not already done so, and try again. The filter doesn't try to install until you try to access a web page from that website, so try that before assuming it hasn't worked.

If the filter is listed, but showing a red cross, then it failed to start. In this case, check the error messages in the event log showing.

If the filter shows as green, with priority unknown, it may not be installed. In this case, you need to check. You can do this with a debug viewer, such as the one supplied by SysInternals (<http://technet.microsoft.com/en-us/sysinternals/bb896647.aspx>). Download this and run it (there's no need to install it) on the server, then restart IIS. If you don't see any messages relating to the PINsafe IIS filter, then it's not installed properly.

If the page is still not redirected, try the following:

1. Check which application pool your web application is running as, then go to the properties page for that application pool.
2. On the Identity tab, change the user to ?Local System?. You will be warned that this is a potential security risk, but don't worry ? it won't be left like this.
3. Restart IIS.
4. Try accessing a protected page again. Hopefully this time you will be redirected.
5. You can now go back to the application pool and change the identity back to what it was originally: it would appear that it is only necessary to run as an administrator to get the filter to register initially.

If you do not see a Turing image when using start session then in a web browser test the following link from the IIS server. If an image is not seen, then there is a problem either with communicating with the PINsafe server or the Allow Image request by username may be set to No.

For an virtual or hardware appliance Install

https://<pinsafe_server_ip>:8443/proxy/SCImage?username=<username>

For a software only install see [Software Only Installation](#)

If the web page is redirected to the /PINsafe/login.asp page but an error message appears then ensure that the Active Server Pages are allowed. To verify this select the Internet Information Services Manager expand the required server then click on Web Service Extension.

16.9.1 Error Messages

AgentXML request failed, error: The agent is not authorised to access the server

User fails to authenticate with the above error message in the PINsafe log. An Agent on PINsafe server has not been defined for the IIS server. Go to Server/Agents in the PINsafe admin console, and add a new entry, using the IP address of the IIS server. Make sure the agent secret is the same as on the IIS filter configuration.

This installation package is not supported on this processor type. Contact your product vendor

The 32 bit version is being attempted to be installed on a 64 bit OS or the 64 bit version is being attempted to be installed on a 32 bit OS. Verify the OS version and install the correct PINsafe software version.

17 Microsoft IIS version 7 ASP.NET Forms Integration

17.1 Introduction

Swivel allows ASP.NET application authentication using Agent-XML for IIS 7 and IIS 6 ASP.NET

NOTE: the method listed here uses standard ASP.Net forms-based authentication to authenticate to PINsafe. We now have an alternative solution that uses a HTTP module. This might be an easier solution than the manual method described below, as all installation and configuration is done using provided applications. Documentation for this solution can be found [here](#).

17.2 Prerequisites

PINsafe

ASP.NET application

ASP.NET Server

17.3 Baseline

PINsafe 3.7

IIS6 and IIS7

17.4 Architecture

The ASP.NET application makes authentication requests against the PINsafe server by Agent-XML.

17.5 ASP.NET Sample Files

ASP.NET Sample File is available here: [ASP.NET Sample File](#)

ASP.NET Sample file for 2008 server is available here: [ASP.NET for 2008 Server](#)

The pinsafe folder contains an example login page, plus aspx pages which render a Turing image or request a dual channel image.

17.6 PINsafe Configuration

17.6.1 Configure a PINsafe Agent

1. On the PINsafe Management Console select Server/Agent
2. Enter a descriptive name for the Agent
3. Enter the IP address or hostname of the server on which the ASP.NET will be running
4. Enter the shared secret used above on the ASP.NET
5. Click on Apply to save changes

Agents: Name:	<input type="text" value="local"/>	
Hostname/IP:	<input type="text" value="127.0.0.1"/>	
Shared secret:	<input type="password" value="....."/>	
Group:	<input type="text" value="---ANY---"/>	
Authentication Modes:	<input type="text" value="ALL"/>	<input type="button" value="Delete"/>
Name:	<input type="text" value="IIS"/>	
Hostname/IP:	<input type="text" value="192.168.1.1"/>	
Shared secret:	<input type="password" value="....."/>	
Group:	<input type="text" value="---ANY---"/>	
Authentication Modes:	<input type="text" value="ALL"/>	<input type="button" value="Delete"/>

Note: Session creation by username is not required for this integration as PINsafe can use session ID.

17.7 ASP.NET Configuration

17.7.1 Integrating the ASP.NET

First of all, extract the sample zip file to a temporary location. There should be 2 folders:

- App_Code
- pinsafe

and one file:

- web.config.

Copy the pinsafe folder and its contents into the ASP.NET application you want to protect or the root of the website to protect the entire website. It is important that the folder is contained within the application, and is not an application in its own right. You will need to set IIS (or other ASP.NET server) to allow anonymous access to the pinsafe folder, and you may need to modify permissions on the files to ensure that the default IIS (or other ASP.NET server) user has read access.

Copy the contents of the App_Code folder into the App_Code folder of the application or create one if it doesn't already have one.

Edit the web.config file for the application, and add the contents of the enclosed web.config in the appropriate locations. You will need to change the PINsafe server settings as appropriate.

17.7.2 Configure the web.config file

This file contains the information for communication with the PINsafe server. The options are displayed below:

PINsafeServer: The IP address or hostname of the PINsafe server or appliance

PINsafePort: The port used for communication, usually 8080

PINsafeContext: The install name of pinsafe, usually pinsafe

PINsafeSecret: The shared secret key, which must be the same as that entered on the PINsafe server

PINsafeSecure: This is if the connection to the PINsafe server is https for SSL or http. The default value is true, which is for https

PINsafePassword: This is to display the password field, the default value of false will not display a password field

PINsafeImage: This is to display a button to generate a Single Channel Image of the security string

PINsafeMessage: This is to display a button to generate a Dual Channel security string to be sent to the user

PINsafeAcceptSelfSigned: If self signed certificates are accepted, default is yes

NOTE: As the requests are made using Agent-XML, they must be made to the pinsafe appliance on port 8080 and the context of pinsafe and not the proxy port of 8443. Security is usually provided by the IIS server proxying the request to the PINsafe server.

Default Settings, suitable for a software install of PINsafe are:

```
<add key="PINsafeServer" value="pinsafe_server" />
<add key="PINsafePort" value="8080" />
<add key="PINsafeContext" value="pinsafe" />
<add key="PINsafeSecret" value="secret" />
<add key="PINsafeSecure" value="true" />
<add key="PINsafePassword" value="false" />
<add key="PINsafeImage" value="true" />
<add key="PINsafeMessage" value="false" />
<add key="PINsafeAcceptSelfSigned" value="true" />
```

Appliance settings are likely to be:

```
<add key="PINsafeServer" value="pinsafe_server" />
<add key="PINsafePort" value="8080" />
<add key="PINsafeContext" value="pinsafe" />
<add key="PINsafeSecret" value="secret" />
<add key="PINsafeSecure" value="true" />
<add key="PINsafePassword" value="false" />
<add key="PINsafeImage" value="true" />
<add key="PINsafeMessage" value="false" />
<add key="PINsafeAcceptSelfSigned" value="true" />
```

17.7.3 Additional web.config file IIS7 Options

The loginUrl setting assumes that you are protecting the entire website. If you are only protecting an application, add the path for that application to this URL. For example, to protect an application with URL "/secure", loginUrl="/secure/pinsafe/Login.aspx".

The <modules> section is not relevant if you are protecting an application that is ASP.NET only. These changes allow ASP.NET authentication to be used for static web pages as well as .aspx pages. This is a new feature of IIS7.

17.7.4 Enabling Authentication

For IIS, open the IIS manager, locate the website or application that you are protecting, and double-click the Authentication icon. Make sure that anonymous authentication is disabled, and that forms authentication is enabled, and the URL is as set earlier. Go to the pinsafe sub-folder, select Authentication under there, and make sure anonymous authentication is enabled (you need to be able to access the login pages anonymously).

17.8 Additional Configuration Options

17.9 Testing

Navigate to the login page. Attempting to login with a correct username and password but no one time code should result in failure. Only when a correct PINsafe one time code is entered should the user be logged in. If the Single Channel button is displayed then an image should appear.

17.10 Troubleshooting

To verify the Single Channel Image works, on the ASP.NET server enter the following into a web browser, which should display a Turing image if the sever is functioning correctly:

For a PINsafe appliance install:

`https://<pinsafe_server_ip>:8080/pinsafe/SCImage?username=test`

For a software only install see [Software Only Installation](#)

17.11 Known Issues and Limitations

Requesting a Security String Index would require modification of the login page for an existing button. See also [Multiple Security Strings How To Guide](#)

17.12 Additional Information

For assistance in the PINsafe installation and configuration please firstly contact your reseller and then email Swivel Secure support at support@swivelsecure.com

18 Microsoft IIS version 7 ASP.NET Integration

18.1 Introduction

This solution uses ASP.Net technology, specifically an HTTP Module, to protect specified web pages using Swivel authentication.

NOTE: the method listed [elsewhere](#) uses standard ASP.Net forms-based authentication to authenticate to PINsafe. The solution described on this page is simpler to install and maintain, but if you are familiar with forms-based authentication and want more control over the look and feel of the login page, you may prefer the alternative solution.

18.2 Prerequisites

PINsafe server version 3.6 or later

ASP.NET application running on Microsoft IIS version 7 (or later). The latest release is compatible with Server 2012 R2 IIS 8.5 and with Server 2016 IIS 10.0. Testing on Windows Server 2019 pending.

Versions: Latest Version 2.3.2.0 available from [here](#). This version fixes several reported vulnerabilities relating to redirecting after login and same-site cookies. It requires Microsoft.Net framework 4.8 or later, and ASP.Net 4.0.

Version 2.2.1.1 available from [here](#). This version is compatible with the Microsoft.Net framework version 4.5 or later, and ASP.Net 4.0.

Version, 2.1.1.1, available from [here](#). This version is compatible with Microsoft.Net framework version 4.0 or later, but does not support TLS versions higher than 1.0, so should only be used in Windows Server 2008 R1, which doesn't have native TLS 1.1/1.2 support.

18.3 Architecture

A HTTP module is installed into a specific ASP.Net application, where it checks all incoming requests. Any request requiring PINsafe authentication will be redirected to the Swivel login page, unless the user has already been authenticated to PINsafe.

18.4 PINsafe Configuration

18.4.1 Configure a PINsafe Agent

1. On the PINsafe Management Console select Server/Agent
2. Enter a descriptive name for the Agent
3. Enter the IP address or hostname of the server on which the ASP.NET will be running
4. Enter the shared secret used above on the ASP.NET
5. Click on Apply to save changes

The screenshot shows the PINsafe Management Console configuration interface. It displays two agent configurations. Each configuration includes fields for Name, Hostname/IP, Shared secret, Group, and Authentication Modes, along with a Delete button.

Agent Name	Hostname/IP	Group	Authentication Modes
local	127.0.0.1	---ANY---	ALL
IIS	192.168.1.1	---ANY---	ALL

Note: Session creation by username is not required for this integration as PINsafe can use session ID.

18.5 Filter Installation

To install the filter, simply run the executable program found in the downloadable zip file. You can generally accept the default recommendations, unless you have reason to change them.

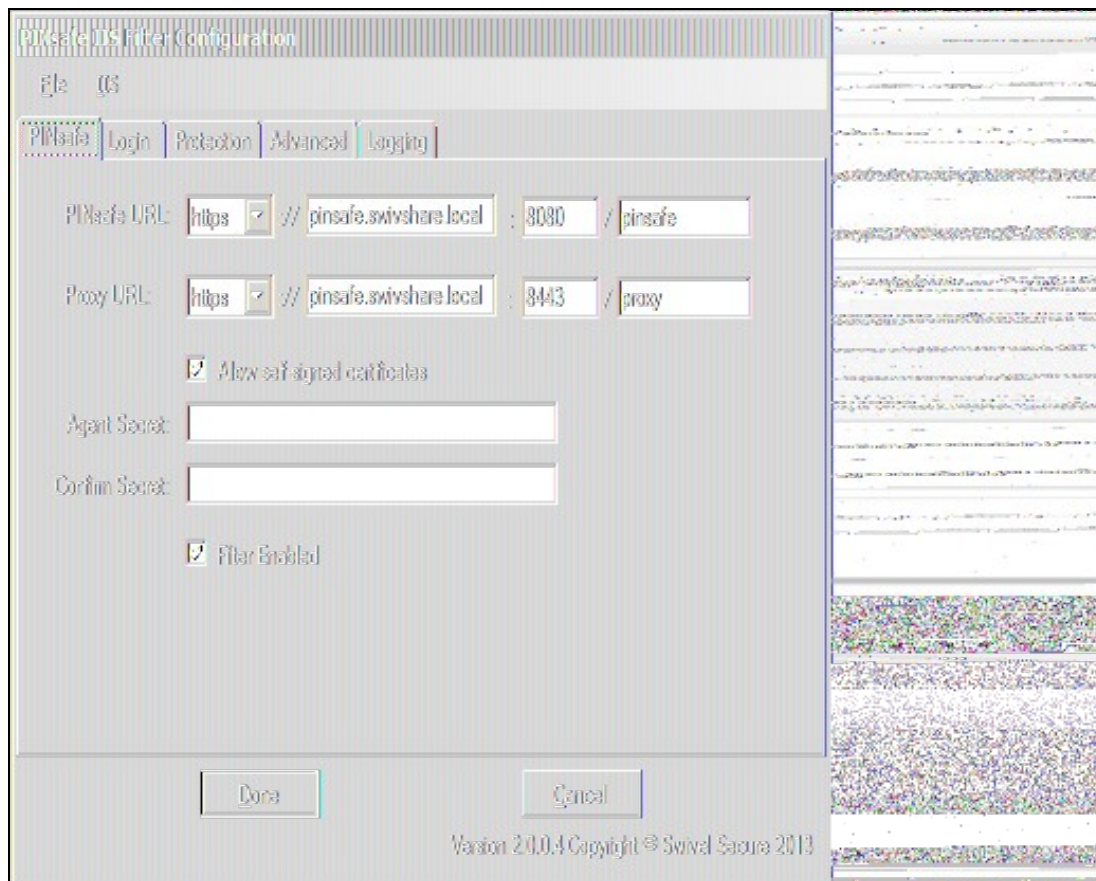
Once the filter is installed, you will be taken to the configuration program (unless you choose not to do so yet).

18.6 Filter Configuration

The filter configuration program enables you to set up which PINsafe server to use for authentication, and also the rules governing which URLs need PINsafe authentication.

The program displays a form with multiple tabs. The tabs are described in separate sections below.

18.6.1 PINsafe Tab



The screenshot shows the 'PINsafe Filter Configuration' dialog box with the 'PINsafe' tab selected. The dialog has a menu bar with 'File' and 'OS'. Below the menu bar are tabs for 'PINsafe', 'Login', 'Protection', 'Advanced', and 'Logging'. The 'PINsafe' tab contains the following fields and options:

- PINsafe URL:** A text field containing 'https://pinsafe.swivshare.local:8080/pinsafe'. The 'https' part is in a dropdown menu.
- Proxy URL:** A text field containing 'https://pinsafe.swivshare.local:8443/proxy'. The 'https' part is in a dropdown menu.
- Allow self signed certificates:** A checked checkbox.
- Agent Secret:** An empty text field.
- Confirm Secret:** An empty text field.
- Filter Enabled:** A checked checkbox.

At the bottom of the dialog are 'Done' and 'Cancel' buttons. The version information 'Version 2.0.0.4 Copyright © Swivel Secure 2013' is displayed at the bottom center.

On this page, you define the PINsafe server settings used for authentication.

Firstly, you define the URL for the PINsafe server, as used to authenticate users.

Secondly, you define the URL for the proxy server, used to deliver single channel images (TURING or PINpad) or dual channel on-demand messages. This may be the same as the PINsafe URL - typically the host name or IP address will be the same. However, if you have an virtual or hardware appliance running PINsafe 3.8 or earlier, PINpad is not available directly from PINsafe. You need to install a recent version of the proxy application, in which case the port and context should be ":8443/proxy", rather than the usual ":8080/pinsafe". These settings will always work for any version of the virtual or hardware appliance. If you have a PINsafe version 3.9 or newer, or are not using PINpad, you can safely use ":8080/pinsafe" for both.

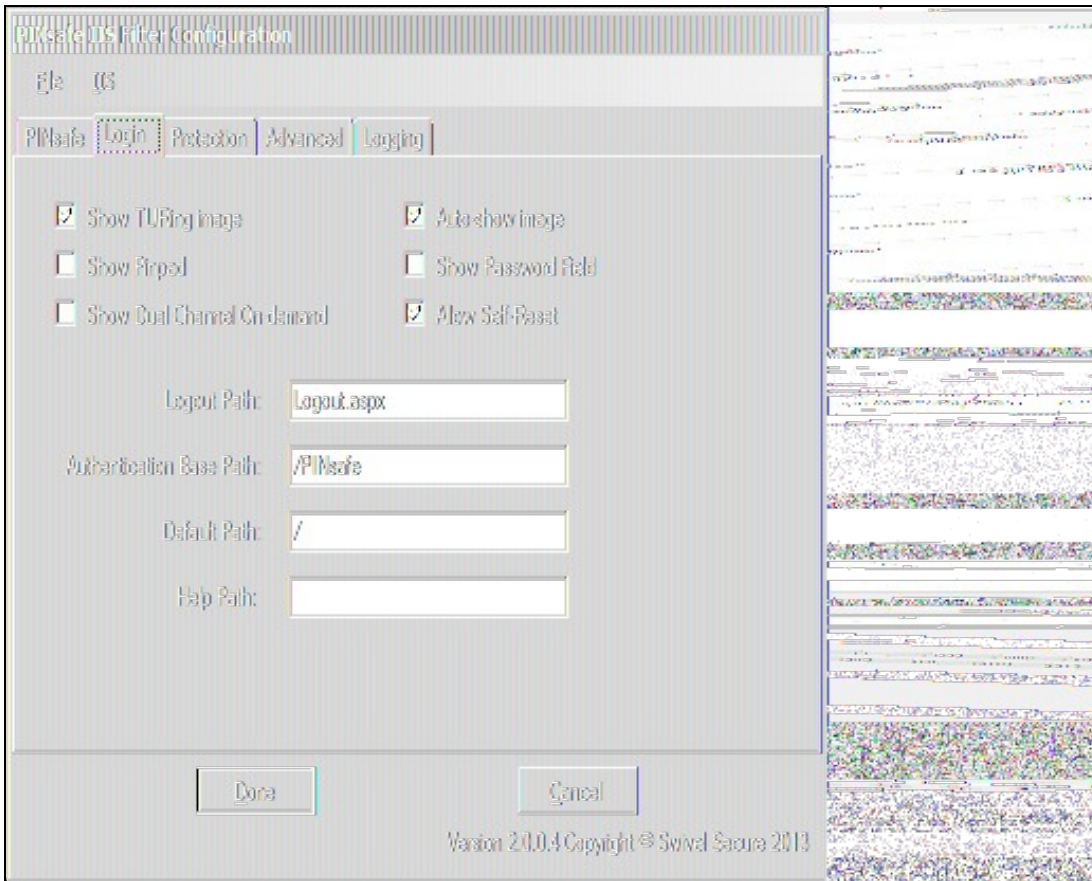
Note that the URLs only need to be resolvable and accessible from the web server. Direct access for the end user to the PINsafe server is not required - the filter proxies all requests.

The next option is "Allow self-signed certificates". If you are using https (recommended), and have specified an IP address for the PINsafe server (not recommended), or have a self-signed or untrusted SSL certificate (not recommended), you need to check this option. For production use, it is recommended that you install a certificate on the Swivel virtual or hardware appliance with the fully-qualified name that you are using to connect to it. If the Swivel virtual or hardware appliance is not visible externally, the certificate can be self-signed or signed by an internal certificate authority, and you can install the signing authority certificate as a trusted certificate on the web server. This is the recommended solution for production use.

Next, you need to enter the Agent secret, which you entered on the PINsafe Agent definition earlier. Enter it twice to confirm it.

The final option on this tab enables or disables the filter. Should you wish to disable the filter temporarily for any reason, you can do this for all websites on this server using this checkbox.

18.6.2 Login Tab



This tab allows you to control the login page used to authenticate to PINsafe.

The 3 checkboxes on the left-hand side allow you to display Turing image, PINpad or a dual-channel on-demand button. You can't have both Turing and PINpad at the same time, but either one can be combined with dual-channel on-demand.

Auto-show image, if checked, will display the Turing image or Pinpad as soon as the username has been entered and the focus moves away from it. This doesn't affect dual-channel on-demand - you always need to click the button for this.

Show Password Field, if checked, will display a password field as well as the OTC field. You only need this if PINsafe passwords are enabled, or the Agent is configured to check the repository password.

Allow self-reset, if checked, will display a link for the self-reset page on the login page. **NOT IMPLEMENTED IN THIS VERSION.**

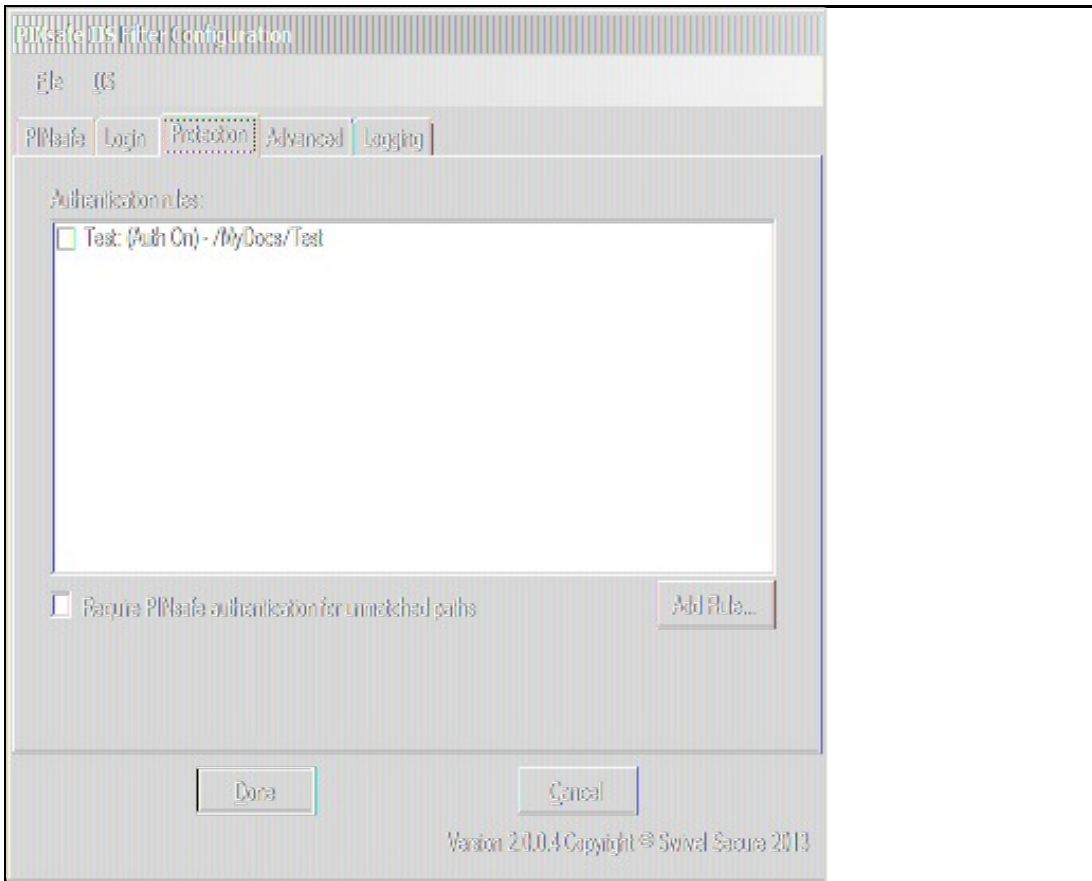
Logout path is the full path used to log out from PINsafe. Typically, this will be /PINsafe/Logout.aspx. If this is detected in the URL, the PINsafe authentication cookie will be removed, and users must re-authenticate to access protected URLs.

Authentication Base Path is the path containing the PINsafe login pages. It will be used when deploying to a web application as the virtual directory. The default is "/PINsafe", and typically you should not need to change this.

Default Path is the path to which the user is redirected after authentication if no source path is provided - for example, if the user navigates directly to the login page. Typically, the user attempts to access a page directly, and is redirected to the login page, with the intended page as the source path.

Help Path is a path to a help file describing how to authenticate to PINsafe. Swivel do not provide such a page, but if the customer wishes to do so, they can enter it here, and a link will be provided on the login page. **NOT IMPLEMENTED IN THIS VERSION.**

18.6.3 Protection Tab

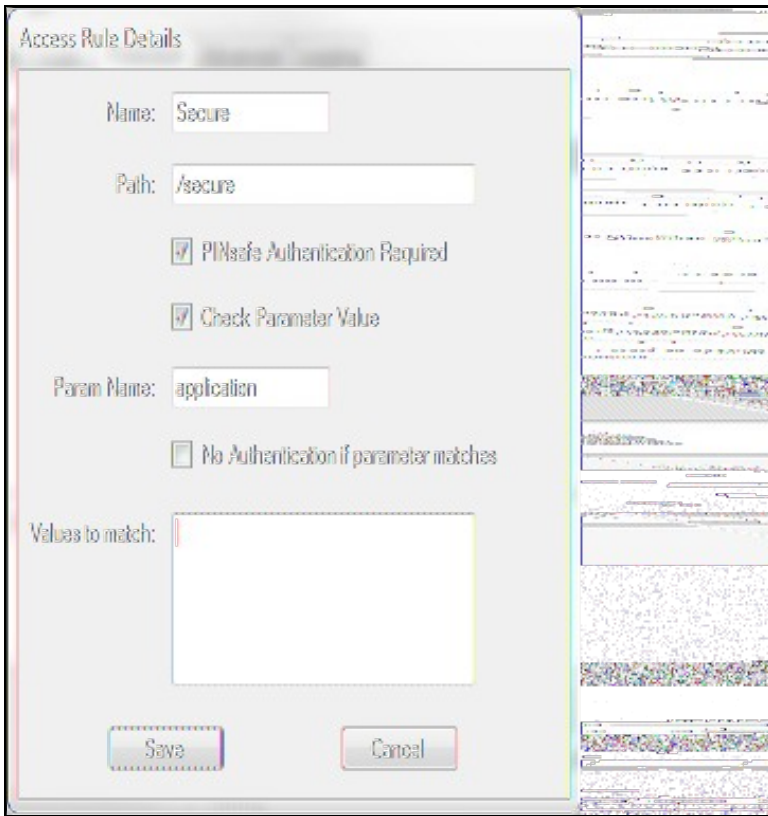


On this page, you specify which paths should require PINsafe authentication. You do this by defining a list of rules. Each rule is a path to be matched, with a flag indicating whether or not PINsafe authentication is required. The filter runs through the rules in order until it matches one, and determines whether or not to check for PINsafe authentication according to that rule.

If no rules match, the default rule can either specify that PINsafe authentication is required or is not required.

NOTE: if you specify the default rule to require PINsafe authentication, make sure that any paths used by the login page are excluded. In particular, you will need a rule for the authentication base path (e.g. "/PINsafe") that does NOT require PINsafe authentication. This is not necessary if the default rule does not require PINsafe authentication.

You can create new rules by clicking the "Add Rule" button. The following dialog appears:



The name is just a label for the rule - it has no intrinsic meaning.

Path is the path that must be matched. By default, the path specified must match the **START** of the request path, so must start with "/": for example, "/secure" will match "/secure/default.aspx" or "/secure/subite/default.aspx", but not "/home/secure/default.aspx". However, if you start the path with a "**", it will match the **END** of the request path: for example "**/default.aspx" will match any page called "default.aspx" anywhere in the website.

"PINsafe Authentication Required" indicates whether or not this rule requires PINsafe authentication.

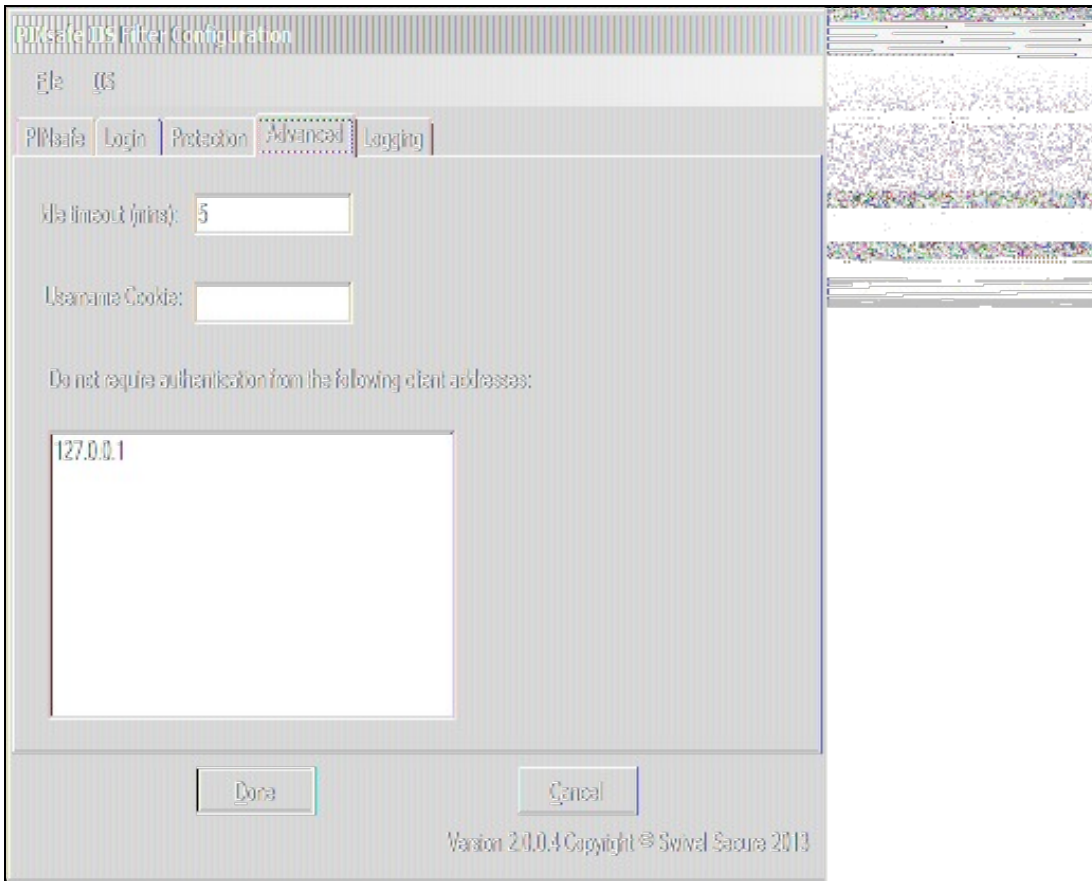
"Check Parameter Value" allows finer control over PINsafe authentication. When checked, you can specify the name of a single query parameter that is checked to determine whether or not PINsafe authentication is required. You can specify a list of possible values for the parameter, but if you specify no values, the presence or absence of the parameter determines whether or not to require authentication.

The final control on this page, "No Authentication if parameter matches", allows you to reverse the parameter check. So for example, if the rule requires authentication, but this option is enabled, PINsafe authentication is required **UNLESS** the parameter value matches one of the specified values.

A final note of clarification: the rule is matched purely on the path, not on the parameters. Specifying "Check Parameter Value" only allows you to change whether or not authentication is required.

Going back to the main form and the list of rules, to change a rule, change the order of rules, or delete rules, check the rules you want to move/change/delete and right-click to bring up a context menu.

18.6.4 Advanced Tab



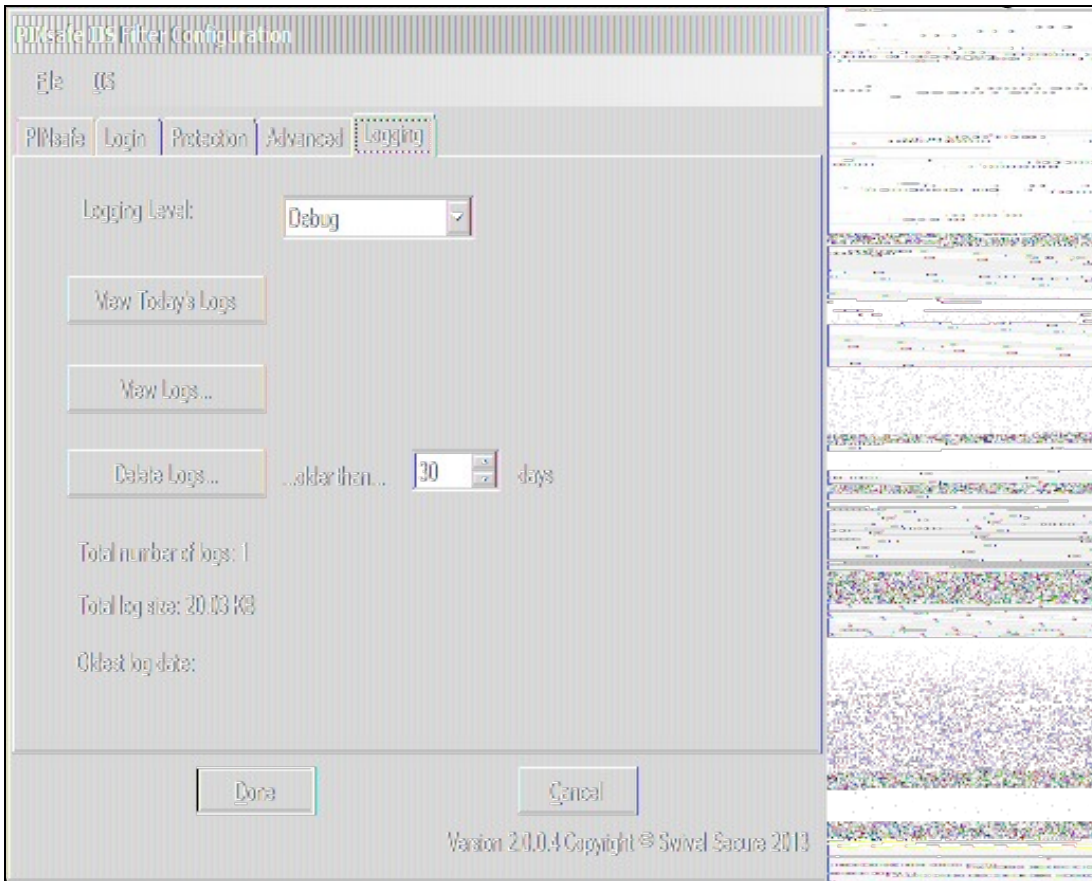
There are 3 settings on this tab:

Idle timeout: this specifies how long the PINsafe authentication cookie is valid if the web page is not refreshed. The default is 5 minutes. If the page is idle for more than 5 minutes, you will need to re-authenticate. You can make this longer if you wish. Note that this doesn't mean that you have to reauthenticate after every 5 minutes - only if you do not refresh the page (or view a different page). Every time a request is made to the website, the timeout resets.

Username cookie: this is provided for additional web development. If you specify a name here, the filter will provide a cookie with the name of the authenticated PINsafe user. **NOT IMPLEMENTED IN THIS VERSION.**

Excluded clients: the final option allows you to specify that PINsafe authentication is not required if the request comes from specified client IP addresses.

18.6.5 Logging Tab

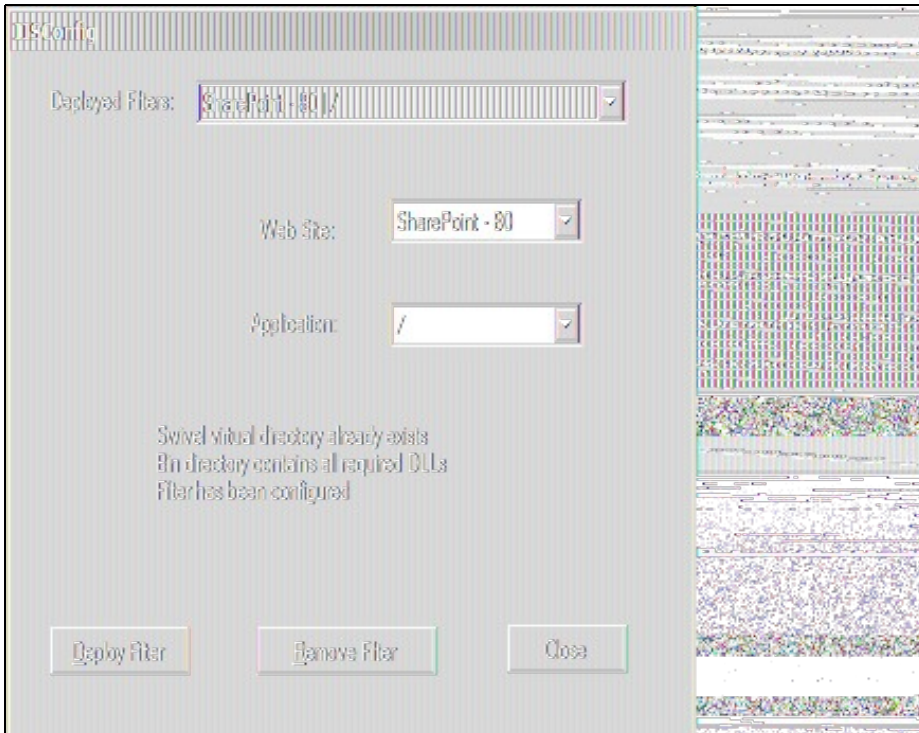


This page allows you to specify what logging the filter does, and to view or delete logs.

There are 4 logging levels: Debug, Info, Error and None. The most verbose, Debug, logs all activity, and all pages checked. Info logs only when a redirect to the login page occurs. Error only logs error events. None disables all logging.

18.6.6 IIS Configuration

None of the option specified above have any effect on any website until the filter is deployed to the website. To do this, Select the IIS menu option, then the Configure sub-menu. The following dialog is displayed:



The first drop-down lists all websites where the filter has been deployed. Initially, therefore, it is empty. If you have already deployed to a website, you can select it to check the status.

The second drop-down lists all websites on the current server. Select one to enable the application drop-down.

The third drop-down list all web applications on the selected website. Select one to check, deploy or remove the filter.

One you have selected a web application, you can choose to deploy or remove the Swivel filter.

18.7 Additional Configuration Options

18.8 Testing

Navigate to the login page. Attempting to login with a correct username and password but no one time code should result in failure. Only when a correct PINsafe one time code is entered should the user be logged in. If the Single Channel button is displayed then an image should appear.

18.9 Troubleshooting

To verify the Single Channel Image works, on the ASP.NET server enter the following into a web browser, which should display a Turing image if the sever is functioning correctly:

For a PINsafe virtual or hardware appliance installs:

`https://<pinsafe_server_ip>:8080/pinsafe/SCImage?username=test`

For a software only install see [Software Only Installation](#)

18.10 Known Issues and Limitations

Requesting a Security String Index would require modification of the login page for an existing button. See also [Multiple Security Strings How To Guide](#)

18.11 Additional Information

For assistance in the PINsafe installation and configuration please firstly contact your reseller and then email Swivel Secure support at support@swivelsecure.com

19 Microsoft IIS version 7 Integration

19.1 Overview

This document outlines the steps required to integrate the Internet Information Server (IIS) with Swivel using dual or single channel authentication. The Swivel install requires configuring an agent on the Swivel server and setting up a shared secret with the IIS server to allow communication for authentication. An ISAPI filter installed on the IIS server allows access to protected resources through the Swivel authentication.

NOTE: This document refers to the version of the filter numbered 1.2, and the configuration application with the same version number. 32-bit and 64-bit versions of the filter are available. Version 1.3.4, with PINpad support, is available for 64-bit only.

If Windows 2008 Server server is being used, or only ASP.Net applications are being protected an alternative authentication is available, see [Microsoft IIS version 7 ASP.NET Integration](#)

19.2 Prerequisites

Internet Information Server on Windows server 2008, 32-bit or 64-bit operating system.

Swivel server

The appropriate Swivel ISAPI filter software can be downloaded from here, depending on your operating system:

The latest release is version 1.3.9. Support for PINpad is included from 1.3.0 onwards. Version 1.3.4 adds PINpad support for change PIN as well:

- [64-bit ISAPI Filter](#)
- [32-bit ISAPI Filter](#)

These links refer to version 1.2 of the filter, provided for legacy purposes.

- [32-bit ISAPI Filter](#)
- [64-bit ISAPI Filter](#)

19.3 IIS Filter Version History

1.2 32 bit and 64 bit

1.3.3 (64-bit only): PINpad support added

1.3.4 (64-bit only): added PINpad support for ChangePIN

1.3.5 (64-bit only): enhancements to ChangePIN support

1.3.6 (64-bit only): added a default logout page

1.3.7-9: various bug fixes

19.4 Swivel Configuration

On the Swivel server configure the agent that is permitted to request authentication. On the Swivel Administration Console select from the server menu Agents and enter the details of the IIS server IP address and a shared key, then click on apply.

Example:

```
Name : IIS server 1,  
Hostname/IP : 192.168.1.1,  
shared secret : secret
```

Agents:	Name:	<input type="text" value="local"/>	
	Hostname/IP:	<input type="text" value="127.0.0.1"/>	
	Shared secret:	<input type="password" value="....."/>	
	Group:	<input type="text" value="---ANY---"/>	
	Authentication Modes:	<input type="text" value="ALL"/>	<input type="button" value="Delete"/>
	Name:	<input type="text" value="IIS"/>	
	Hostname/IP:	<input type="text" value="192.168.1.1"/>	
	Shared secret:	<input type="password" value="....."/>	
	Group:	<input type="text" value="---ANY---"/>	
	Authentication Modes:	<input type="text" value="ALL"/>	<input type="button" value="Delete"/>

If Single Channel communication is to be used, select from the Swivel Administration Console Single Channel, and set the Allow image request by username to Yes then click on apply.

Server>Single Channel

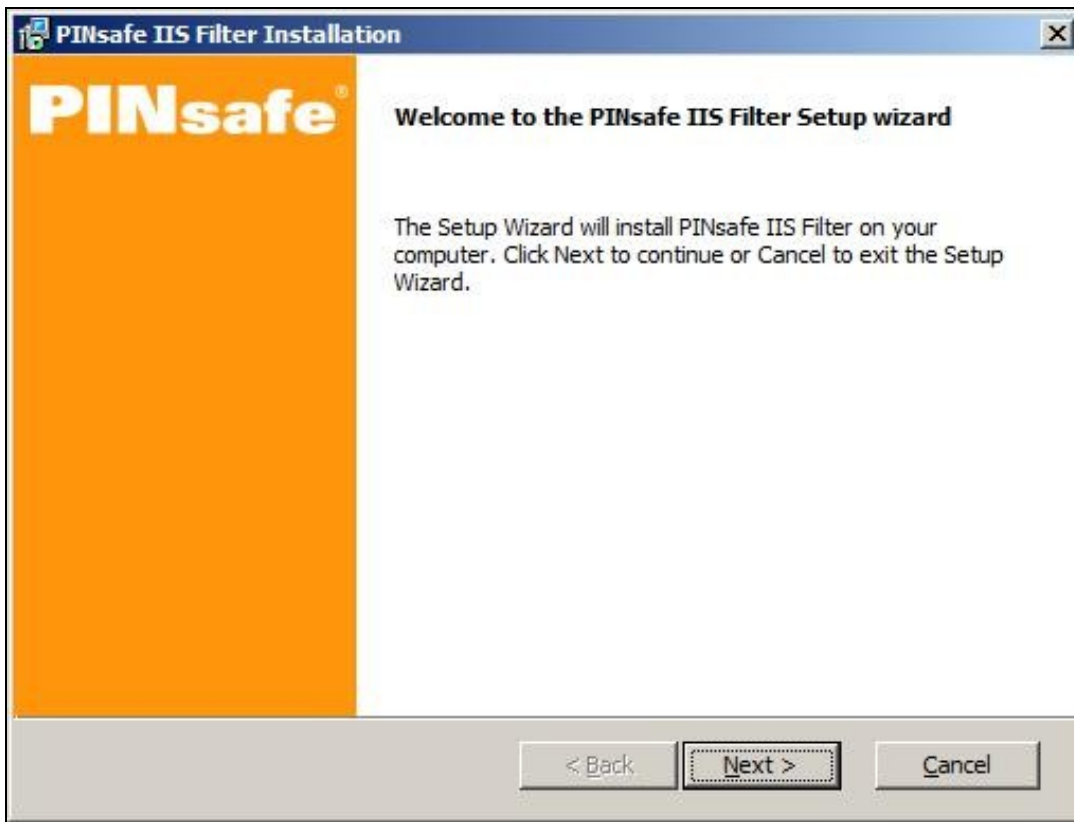
Please specify how single channel security strings are delivered.

Image file:	<input type="text" value="turing.xml"/>
Rotate letters:	<input type="text" value="No"/>
Allow session request by username:	<input type="text" value="Yes"/>
Only use one font per image:	<input type="text" value="Yes"/>
Jiggle characters within slot:	<input type="text" value="No"/>
Add blank trailer frame to animated images:	<input type="text" value="Yes"/>
Text Alpha Value:	<input type="text" value="80"/>
Number of complete display cycles per image:	<input type="text" value="10"/>
Inter-frame delay (1/100s):	<input type="text" value="40"/>
Image Rendering:	<input type="text" value="Static"/>
Multiple AUTHentications per String:	<input type="text" value="No"/>
Generate animated images:	<input type="text" value="No"/>
Random glyph order when animating:	<input type="text" value="No"/>
No. Characters Visible:	<input type="text" value="1"/>

19.5 Configuring the IIS Server

19.5.1 Install the Swivel Filter

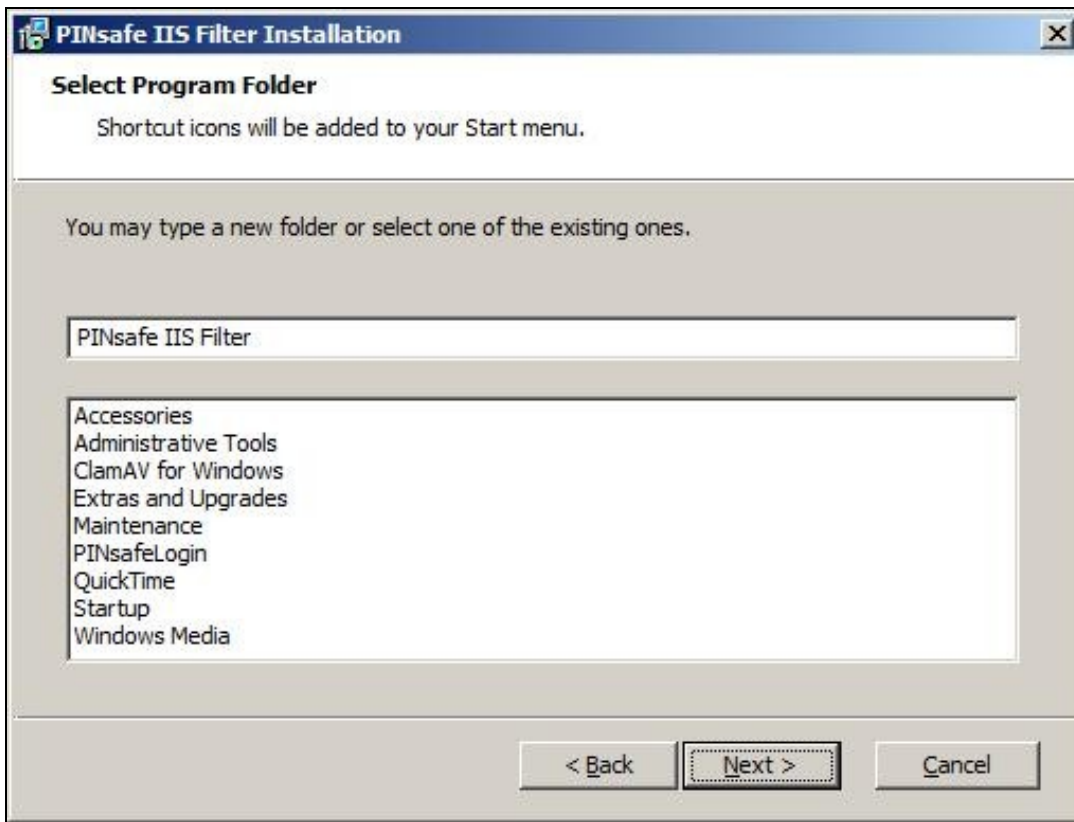
1. On the IIS server run the PINsafellISFilter.exe. The IIS filter may need to be run as an Administrator user (The filter needs to be installed by IIS running as Administrator user but it can run as a normal user).



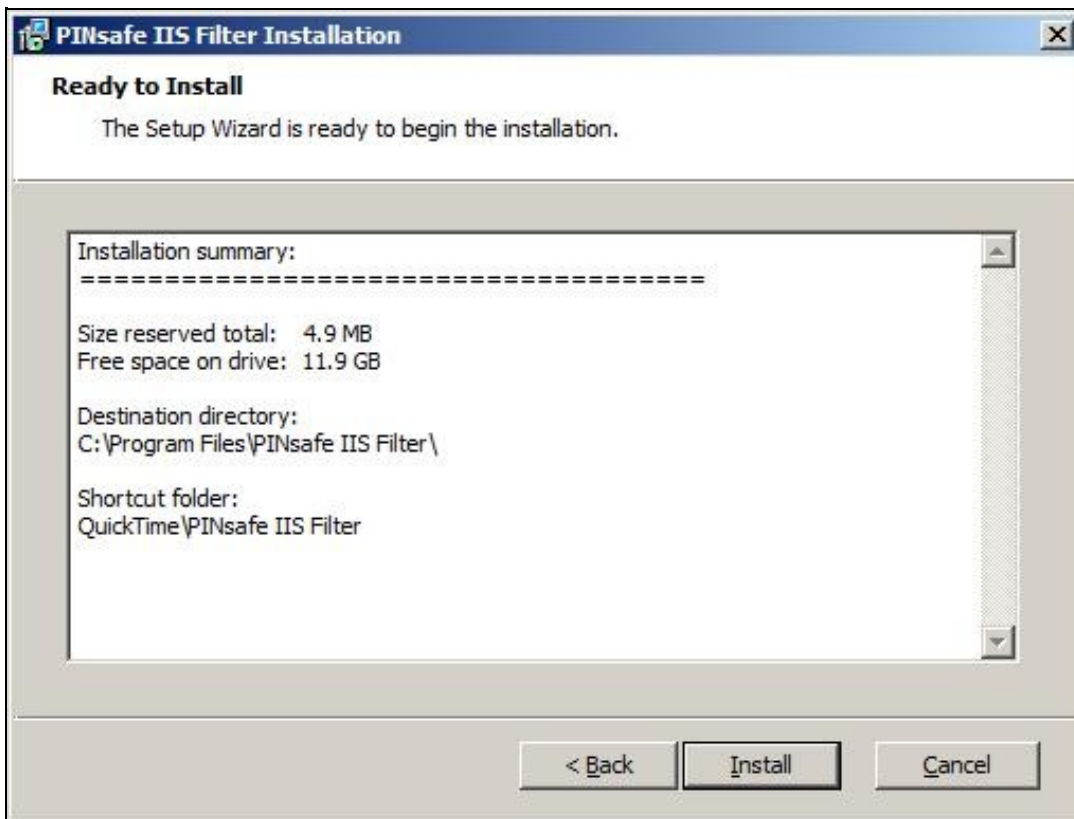
2. Choose the Path to Install to such as C:\Program Files\PINsafe IIS Filter



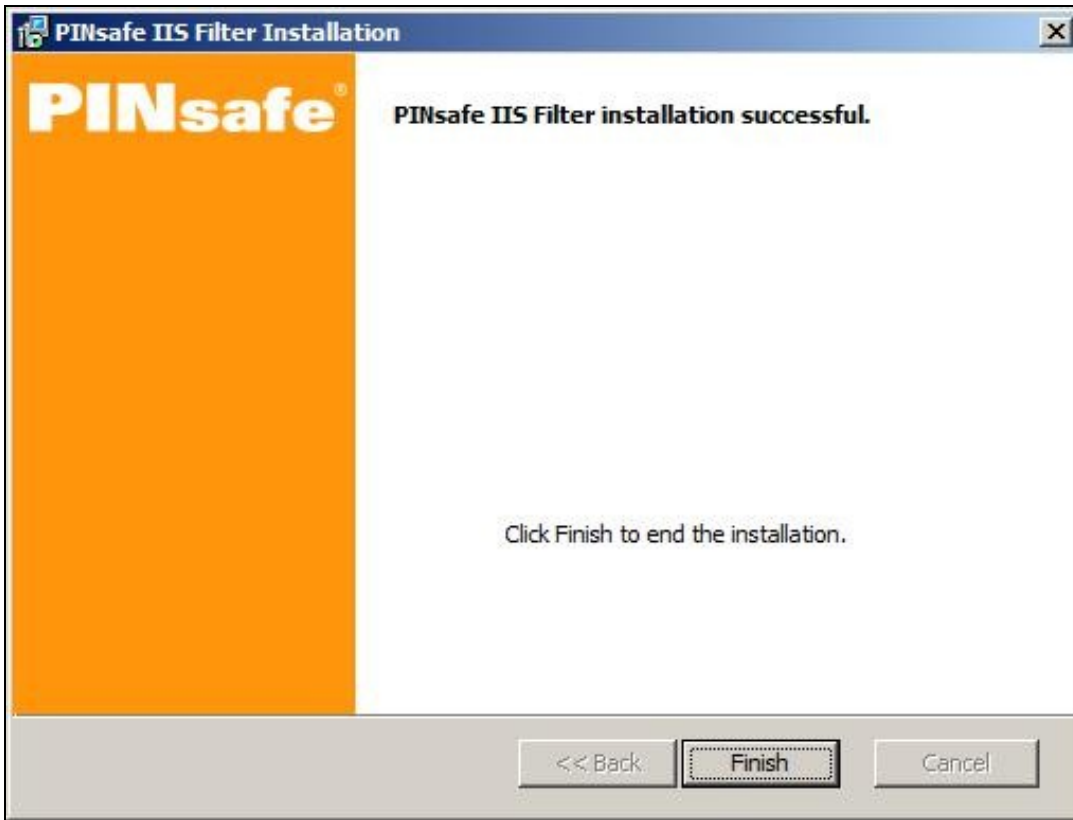
3. Select Start Menu Folder



4. When details are correct click on Install

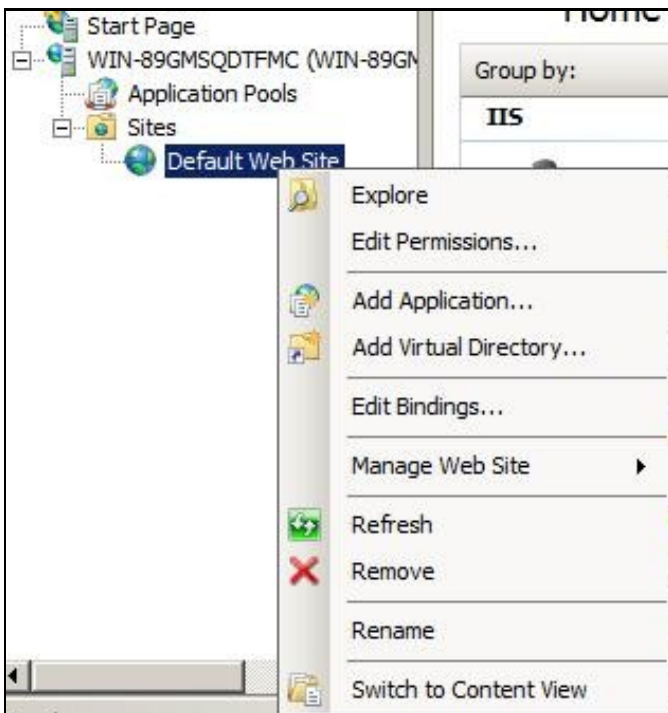


5. If the error ?Incorrect Command Line Parameters? is seen click on OK

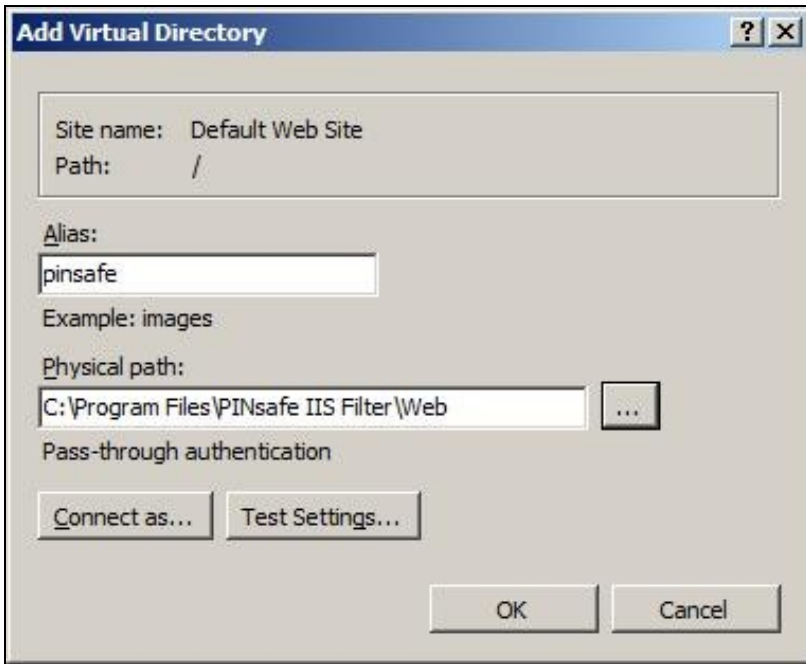


Create a PINsafe virtual directory

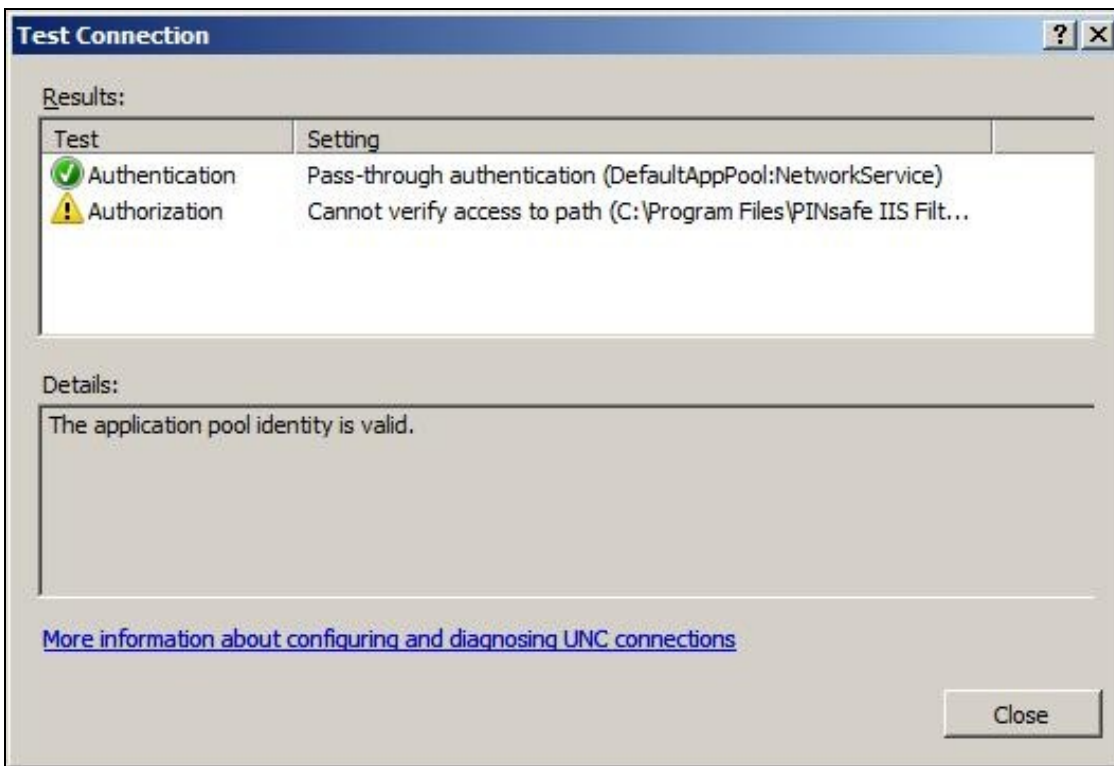
1. On the Internet Information Services Manager right click on the website and select Add Virtual Directory



2. Create an Alias called PINsafe



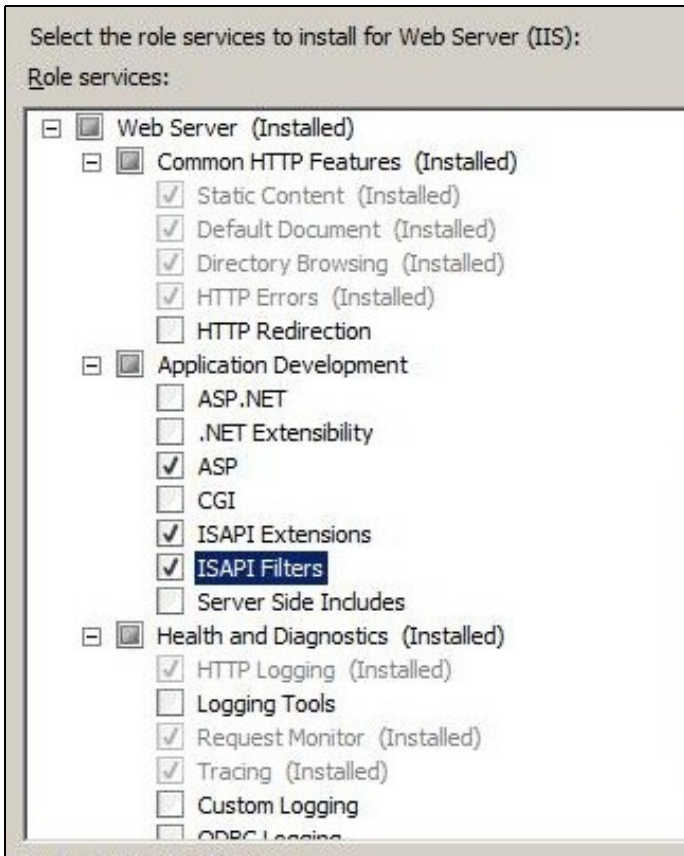
3. Point the path to the PINsafe directory Web folder, by default C:\Program Files\PINsafe IIS Filter\Web. Test Connection verifies the path, and Connect As allows Application User for pass through authentication.



4. Set the permissions to Read and Run Scripts

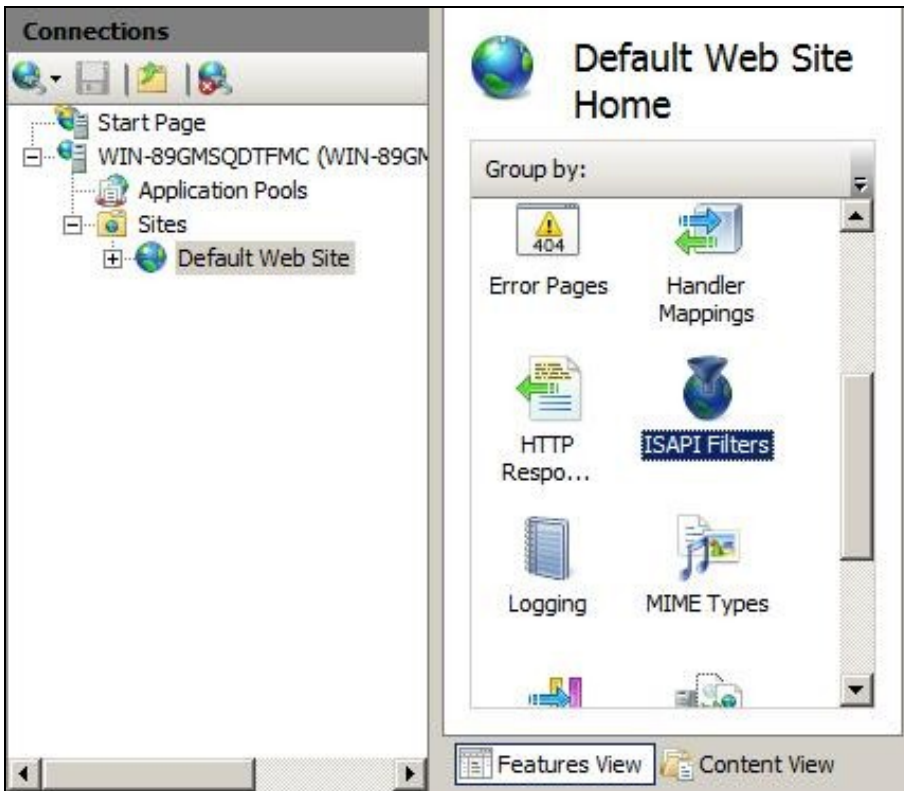
19.5.2 Installing the ISAPI Filters, extensions and ASP on IIS

This requires the ISAPI filters, ISAPI extensions and ASP to be installed. To verify or install these, for Windows 2008, on the IIS server start the Server Manager by selecting Start/Administrative Tools/Server Manager then expand the tab for Roles, click on the Web Server (IIS), then look under Role Services to ensure that the *ISAPI Filters*, *ISAPI Extensions* and ASP are installed. If it is not click on Add Role Services and add them.

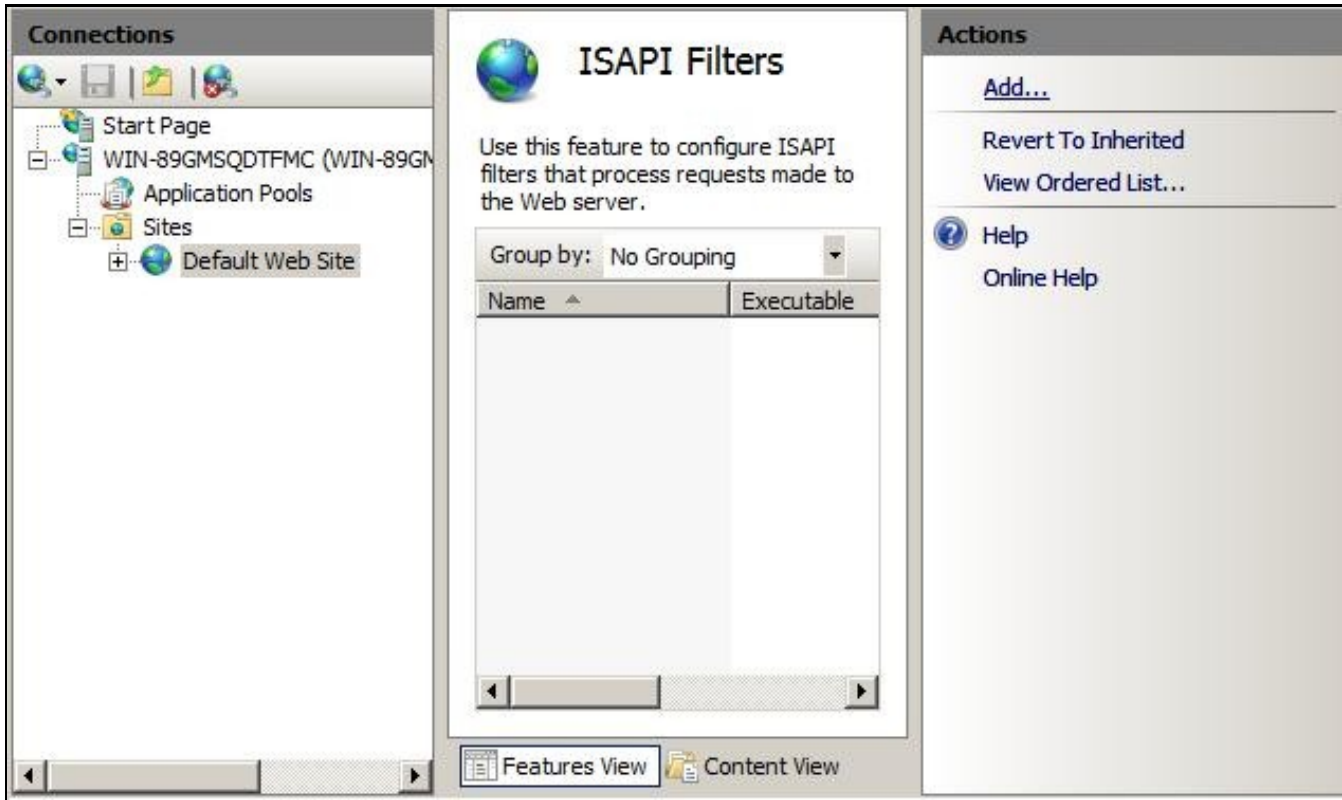


19.5.3 Install the Swivel ISAPI Filter

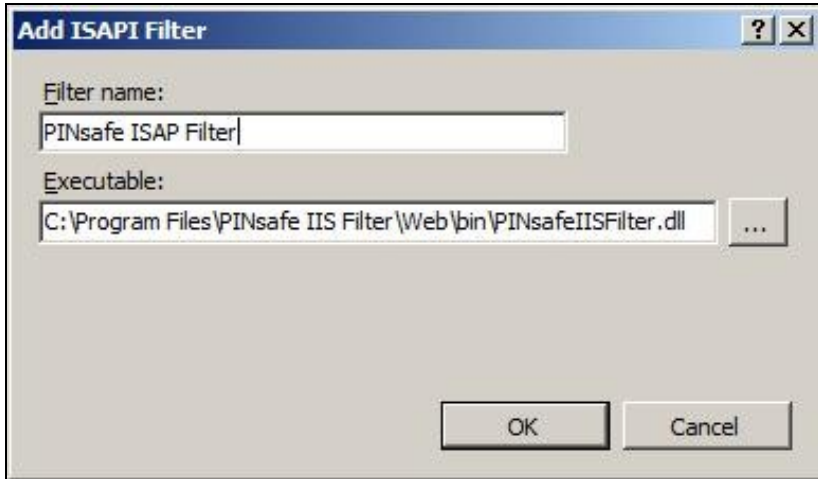
1. On the Internet Information Services Manager Select the website
2. Select ISAPI filters by double clicking on the ISAPI filters icon



3. Under Actions select Add



4. Select the Path to the Swivel ISAPI filter. Note that the actual file you require will be PINsafeIISFilter.dll, located in the sub-folder Web\bin of the installation folder. Enter a name for the Filter such as *PINsafe ISAPI Filter*. When information is complete click on Ok.



5. Ensure the Swivel ISAPI filter is the top filter by selecting the 'View Ordered List...'



19.5.4 Configure the ISAPI filter

1. Select Start Programs/PINsafe IIS Filter/Filter Configuration. This will look for the location of config.xml, this will be created when first used and this must be located in web/bin.

Note: If the Swivel Filter Configuration does not exist in the Start Menu, it can be started by running it from its install location. The default install location is C:\Program Files\PINsafe IIS Filter\Web\bin\ConfigApp.exe

2. You will be asked to select the location of the configuration file. It is important that you select the right location. It should be in the same folder as the ISAPI filter. Initially, this file will not exist, but will be created as a result of running this configuration application.

PINsafellSFilter Options

PINsafeServer: The PINsafe Server tab contains settings which define the Swivel server which will be used to authenticate users.

Hostname/IP: The name or IP address of the Swivel server.

Port: The port number used by the Swivel server (normally 8080, or 8443 for HTTPS).

Context: The context (i.e. web application name) of the Swivel instance on that server

Secret: The common secret used to communicate with the Swivel server. This value must be the same as the secret defined for the Swivel agent configured earlier.

SSL enabled: Tick this box to require SSL (HTTPS) communication with the Swivel server.

Permit self-signed certificates: Tick this box to allow SSL certificates to be self-signed. This also ignores other certificate errors, such as site names not matching.

Authentication: The Authentication tab contains the following settings:

Idle time (s): The time (in seconds) for which the authentication cookie will be valid if the web page is not used.

Username header: The name of a cookie which will pass the username of the authenticated Swivel user. If this value is blank, no cookie will be provided.

Single: Indicates that single channel security strings (i.e. **TURing** image) are permitted.

Dual: Indicates that dual channel security strings (i.e. via e-mail, SMS) are permitted.

On-demand dual: Indicates that the login page should display a button to request dual-channel security strings.

Display password fields: Indicates that the login page should show a field for Swivel password as well as OTC.

Permit self-reset: Indicates that the user self-reset page should be enabled.

Exclusions and Inclusions: Use the inclusion and exclusion tabs to enter which paths should be protected by Swivel:

Included paths: This is a list of paths within the current website which require Swivel authentication. If this list is empty, the entire website will be protected except as indicated by the Exclusions tab. Paths should be one per line.

Excluded paths: This is a list of paths within the current website which should be exempt from Swivel authentication. This is only relevant if the included paths list is empty, in which case all paths not on this list will be protected by Swivel.

Excluded addresses: This is a list of IP addresses which are exempt from Swivel authentication. All requests from these addresses are passed through without authentication.

Misc: On the Misc tab, edit any custom paths as follows:

Default path: This is the path to which authenticated requests are directed if the login page is targeted directly. If a user tries to access a protected page, she is redirected to the login page, and after authentication, back to the page she was trying to access. If the user requests the login page directly, she will be redirected to this location after authentication.

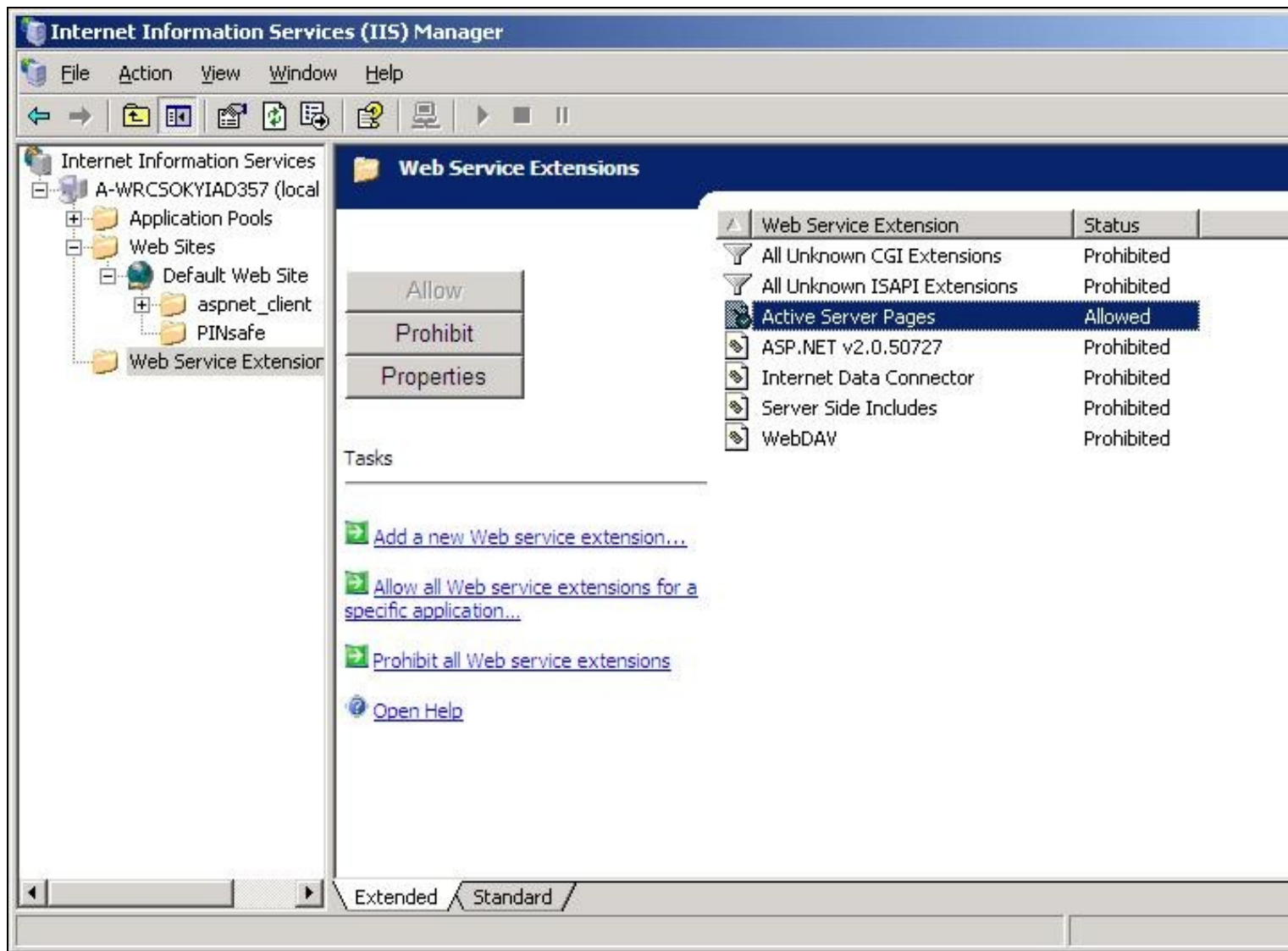
Logout path: Requesting this path will result in the user being logged out. Subsequent requests will require re-authentication, if relevant. If this path is empty, users can only be logged out by closing the browser, or if the authentication times out. Note: The logout path must be an included file location.

Virtual web path: This is the path to the Swivel authentication pages. See the next section for details on setting this up. You should normally set this to be `/?pinsafe?`, unless you have a particular reason not to.

Help URL: The URL for Swivel IIS filter help. The filter does not come with help pages as standard, so this should only be filled in if help pages have been provided by the reseller or end user.

After configuration is complete Apply the settings and restart the World Wide Web Publishing Services.

Allow Active Server Pages: to verify this, select the Internet Information Services Manager expand the required server then click on Web Service Extension.



19.6 Installing the Filter on Multiple Websites

Most of the instructions above assume that you are installing the IIS filter on the single default website of a web server. However, if you want to use the filter on multiple websites on the same server, you need to carry out a few extra steps.

Firstly, if all the settings on all the websites are exactly the same, you can configure it once, and then carry out the same steps to activate the filter on each website.

If, however, you need different settings for each website, you will need to do the following for all except the first website:

1. Make a copy of the entire Web folder, including the bin sub-folder. You can copy it as a sub-folder of Swivel IIS filter, with a different name, or you can put it somewhere completely different. It is important, however, that you keep the structure the same - all the .asp files must remain intact, and the filter DLL must be in a sub-folder called 'bin'.
2. When you come to configure each filter, navigate to the bin sub-folder of the copy you have just made. Otherwise, configuration is exactly as before.
3. When selecting the IIS filter to install, and also when defining the virtual directory for Swivel web pages, you should use the copy you have just created, rather than the original. This is important, as both the filter and the web pages look for the configuration file relative to their own location.

19.7 Testing

Browse to a web page that has been configured for protection. This should display a Swivel login dialogue:



A login form with three input fields: Username, Password, and OTC. Below the fields are two buttons: Start Session and Login.

Enter the Username.

For dual channel, enter the One Time Code:



The login form with 'admin' entered in the Username field and five dots in the OTC field. The Login button is highlighted in blue.

Or click start session to enter a single channel OTC. The Swivel log will record that a single channel session has started.



The login form with 'admin' entered in the Username field and five dots in the OTC field. The Start Session button is highlighted in blue.

If authentication is successful it should redirect to the login page. If failed an error message will appear. The Swivel log will record any successful log attempt for the agent.



The login form with 'admin' in the Username field and empty Password and OTC fields. The Start Session button is highlighted in blue. An orange error message box is displayed at the bottom.

An error occurred, please check your credentials. If the error persists contact your PINsafe Administrator.

19.8 Uninstalling the filter

To remove the Filter, remove role services that are not required by other applications, to do this for Windows 2008, on the IIS server start the Server Manager by selecting Start/Administrative Tools/Server Manager then expand the tab for Roles, click on the Web Server (IIS), then look under Role Services to remove the *ISAPI Filters*, *ISAPI Extensions* and ASP are installed. The system will require a restart to complete.

From the IIS Manager right click on the Swivel Virtual Directory, then select Remove, Click on Yes to Confirm.

To uninstall the Swivel IIS Filter, choose Start/All Programs/PINsafe IIS Filter/PINsafe IIS Filter Uninstaller, then click Yes on the confirmation to uninstall.

The Swivel Filter config may be left after uninstalling, so to completely remove this, remove the folder Program Files\PINsafe IIS Filter.

19.9 Troubleshooting

Verify the correct filter version has been installed for the operating system i.e. 32 bit or 64 bit.

Check for error messages in the Swivel log

Check the IIS log messages

Filter Install issues: The IIS service needs to run as an administrator in order to install the filter. Once the filter is installed, it will run as the normal user. Open the Services applet and locate World Wide Web Publishing. Select properties, then go to the Log On tab. Assuming you are logged in as an administrator, select Local System account. Then restart the service. Hopefully, this will start the filter. You can then switch back to the default account and restart again.

If you are not redirected to the Swivel login page when trying to access a protected page, open IIS manager and check that the ISAPI filter is installed and has loaded properly (there should be a green arrow next to it, and the priority should be 'High?'). If this is not the case, restart IIS, if you have not already done so, and try again. The filter doesn't try to install until you try to access a web page from that website, so try that before assuming it hasn't worked.

If the filter is listed, but showing a red cross, then it failed to start. In this case, check the error messages in the event log showing.

If the filter shows as green, with priority unknown, it may not be installed. In this case, you need to check. You can do this with a debug viewer, such as the one supplied by SysInternals (<http://technet.microsoft.com/en-us/sysinternals/bb896647.aspx>). Download this and run it (there's no need to install it) on the server, then restart IIS. If you don't see any messages relating to the Swivel IIS filter, then it's not installed properly.

If the page is still not redirected, try the following:

1. Check which application pool your web application is running as, then go to the properties page for that application pool.
2. On the Identity tab, change the user to 'Local System?'. You will be warned that this is a potential security risk, but don't worry ? it won't be left like this.
3. Restart IIS.
4. Try accessing a protected page again. Hopefully this time you will be redirected.
5. You can now go back to the application pool and change the identity back to what it was originally: it would appear that it is only necessary to run as an administrator to get the filter to register initially.

If you do not see a Turing image when using start session then in a web browser test the following link from the IIS server. If an image is not seen, then there is a problem either with communicating with the Swivel server or the Allow Image request by username may be set to No.

For a virtual or hardware appliance Install

`https://<pinsafe_server_ip>:8443/proxy/SCImage?username=<username>`

For a software only install see [Software Only Installation](#)

If the web page is redirected to the /PINsafe/login.asp page but an error message appears then ensure that the Active Server Pages are allowed. To verify this select the Internet Information Services Manager expand the required server then click on Web Service Extension.

No authentication on main page

Open IIS Manager and disable Anonymous authentication for the root folder. Refresh the browser to prevent caching and try again.

You may need to ensure that Anonymous authentication is enabled for the PINsafe folder, though, so you don't run into problems showing the TURING image.

Authentication working internally but not externally

If it is working internally, but not externally, ensure that there is no caching by opening a new browser. Also specify the default redirect URL as "/default.htm", rather than "./default.htm". The latter will redirect to default.htm within the pinsafe folder.

19.9.1 Error Messages

AgentXML request failed, error: The agent is not authorised to access the server

User fails to authenticate with the above error message in the Swivel log. An Agent on Swivel server has not been defined for the IIS server. Go to Server/Agents in the Swivel admin console, and add a new entry, using the IP address of the IIS server. Make sure the agent secret is the same as on the IIS filter configuration.

This installation package is not supported on this processor type. Contact your product vendor

20 Microsoft RD Web Access

21 Introduction

This filter allows you to protect Windows Remote Desktop Services (RDS) Web Access with Swivel authentication.



MS RD Web & TURING



MS RD Web & SMS / Mobile App.

22 Prerequisites

Swivel version 3.x or 4.x

Windows Server 2012 R2 or Windows Server 2016 with RDS Web Access already installed

Microsoft.Net Framework version 4.5, full edition (rather than client-only) installed

A version compatible with Windows Server 2008 is also available. This requires Microsoft.Net framework 4.0 only.

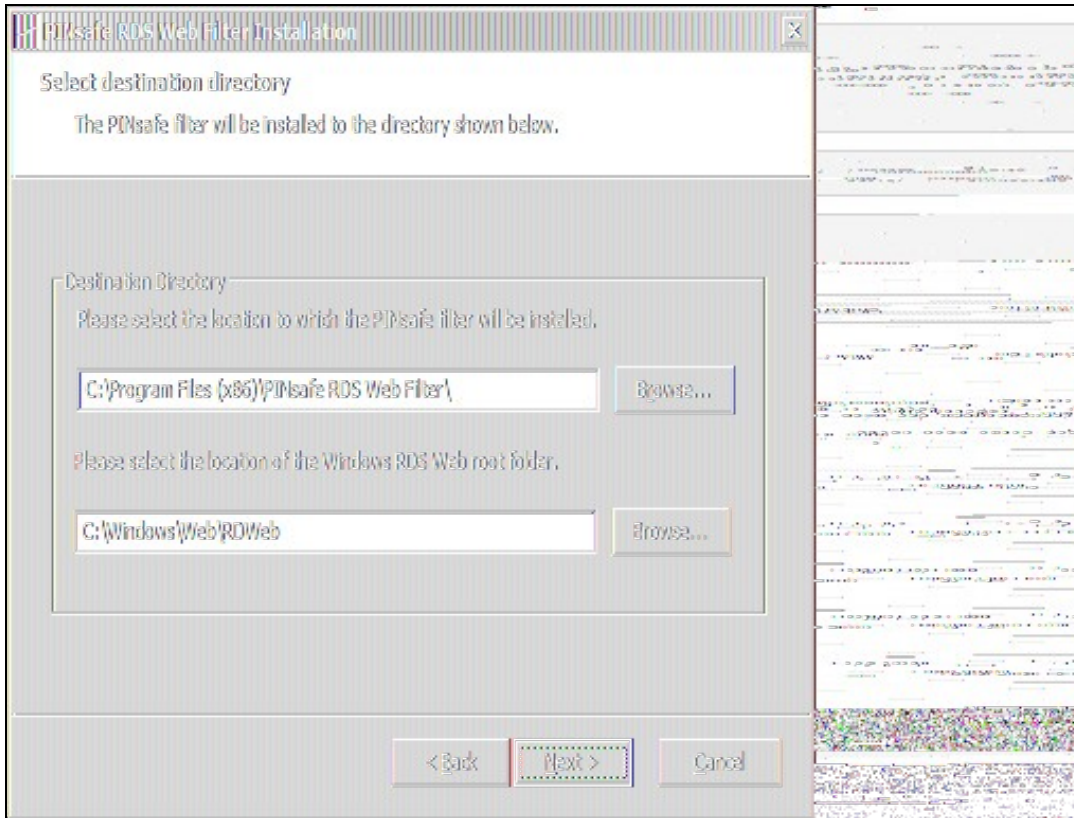
23 Swivel Server Configuration

The only configuration you need to do on the Swivel server is to ensure that the RDS server is configured as an Agent for Swivel (under Server -> Agents), and if you are using the TURing image or PINpad, that under Server -> Single Channel, the option Allow session request by username is set to Yes.

24 Installation

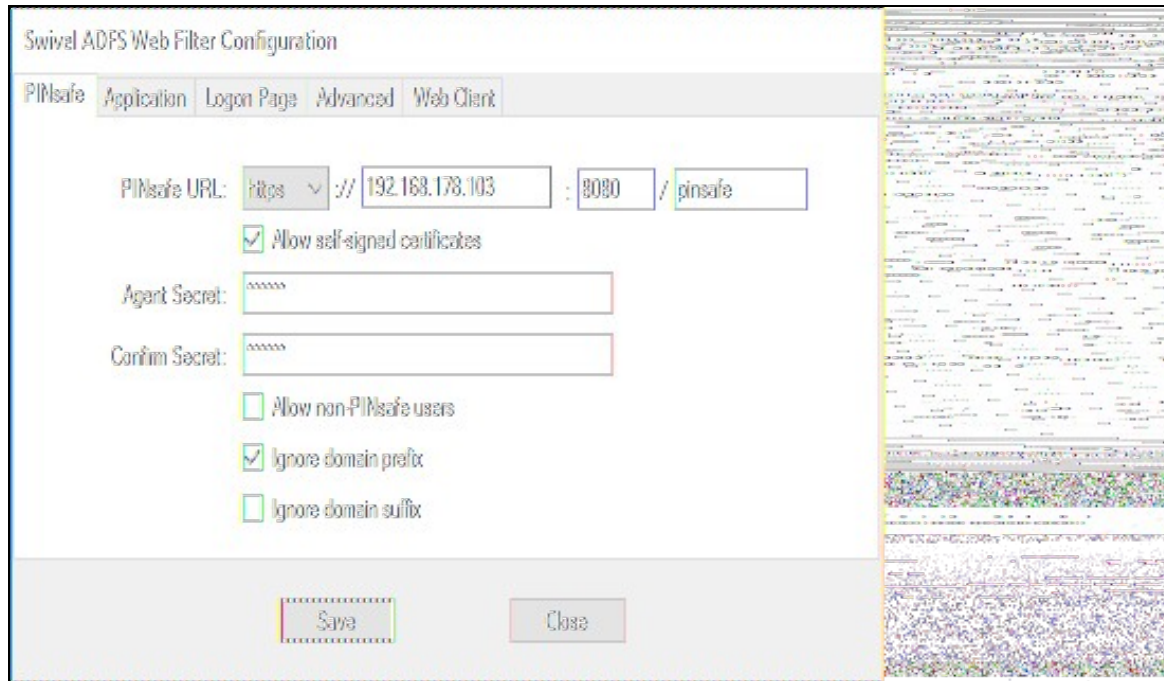
You can download the Windows Server 2019 filter from [here](#), the Windows Server 2016 filter from [here](#) and the Windows Server 2012 R2 filter from [here](#). The version compatible with Windows Server 2008 is available from [here](#).

Installation consists of a single executable, RDSWebFilterInstaller.exe. In most cases you can accept the default settings during installation. When you get to the destination folder, make sure that the RDS web root folder is selected correctly. In most cases, C:\Windows\Web\RDWeb will be correct, but make sure if your configuration is not a default installation that the right folder is selected.



25 Configuration

When installation is completed, you will be presented by the configuration page, as shown here.



The screenshot shows the 'Swivel ADFS Web Filter Configuration' dialog box with the 'PINsafe' tab selected. The configuration fields are as follows:

- PINsafe URL:** https://192.168.178.103:8080/pinsafe
- Allow self-signed certificates
- Agent Secret:** [Redacted]
- Confirm Secret:** [Redacted]
- Allow non-PINsafe users
- Ignore domain prefix
- Ignore domain suffix

Buttons at the bottom: Save, Close.

25.1 Configuration Options

PINsafe URL: select https or http, enter the Swivel IP or hostname. Use port 8080, unless you have a custom installation. The context will be "pinsafe" for version 3.x and "sentry" for version 4.x.

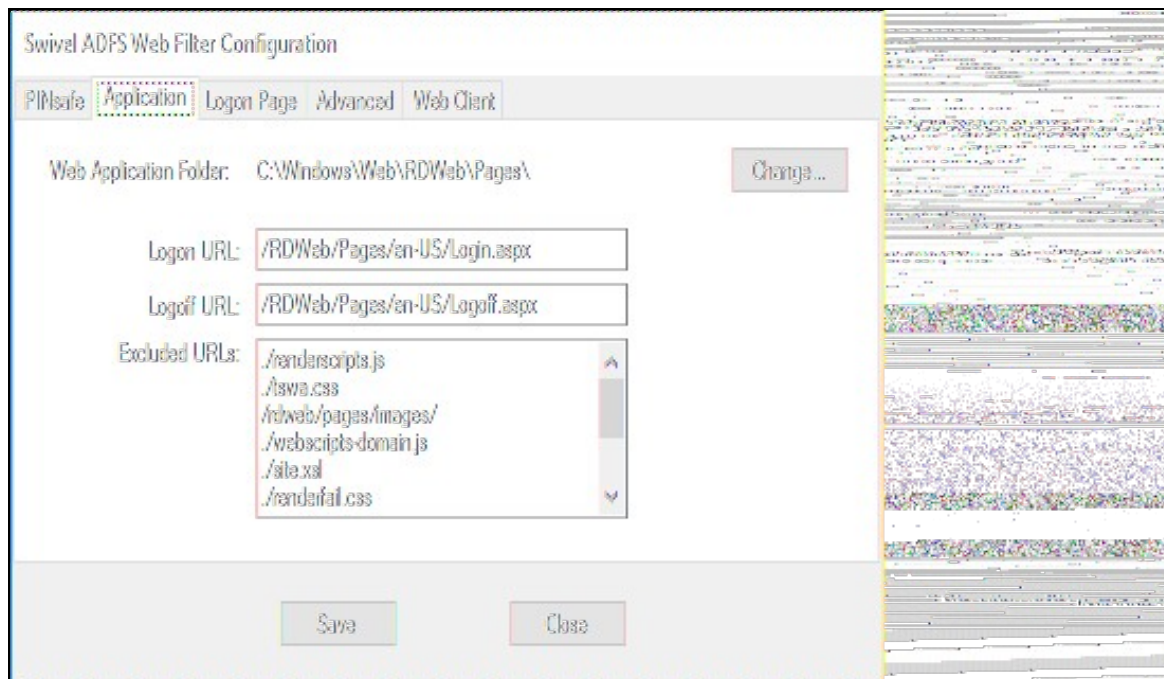
Note: do not use the ?:8443/proxy? URL, as that is not valid for authentication.

Allow self-signed certificates Check box, Check the box to ignore certificate errors

Agent Secret: and **Confirm Secret:** The shared secret entered on the Swivel instance under Server/Agents

Allow non-PINsafe Users if checked permits users that do not have PINsafe accounts to log in with just username and password.

Ignore domain prefix and **Ignore domain suffix** if checked remove the domain name before or after the username before passing to PINsafe. The fully-qualified name is always passed to Windows for authentication.



The screenshot shows the 'Swivel ADFS Web Filter Configuration' dialog box with the 'Application' tab selected. The configuration fields are as follows:

- Web Application Folder:** C:\Windows\Web\RDWeb\Pages\ (Change... button)
- Logon URL:** /RDWeb/Pages/en-US/Login.aspx
- Logout URL:** /RDWeb/Pages/en-US/Logout.aspx
- Excluded URLs:**
 - ./renderscripts.js
 - ./swa.css
 - /rdweb/pages/images/
 - ./webscripts-domain.js
 - ./site.xml
 - ./renderfail.css

Buttons at the bottom: Save, Close.

Web Application Folder: Change allows a new path to be specified

The following settings you will probably not need to change, unless you have customised your login page. In this case, make sure that any images, scripts or stylesheets you have added are listed under the Excluded URLs. An entry beginning with ?./? will match any path that ends with the remaining part of the path: for example, ?./renderscripts.js? will match the file renderscripts.js wherever it is in the web hierarchy. Any files not listed under Excluded URLs, or the logon or logoff path, will be blocked by the Swivel filter, until you have authenticated to Swivel.

Logon URL: default: /RDWeb/Pages/en-US/Login.aspx

Logoff URL: default: /RDWEB/Pages/en-US/Logoff.aspx

Excluded URLs: list of URLs for which authentication is excluded. NOTE: URLs must be entered one per line, but unfortunately, it is not possible to enter new lines into this box. To change it, you must therefore copy the current list into a text editor, make any changes required and then paste the new list back.

The screenshot shows the 'Swivel ADPS Web Filter Configuration' dialog box with the 'Logon Page' tab selected. The dialog has a title bar and a menu bar with 'PINsafe', 'Application', 'Logon Page', 'Advanced', and 'Web Client'. The main area contains several checkboxes and text input fields. The checkboxes are: 'Show TURing image' (checked), 'Show Request String' (unchecked), 'Show Pinpad' (unchecked), 'Show blank image for unknown user' (unchecked), 'Auto-display image' (checked), and 'Auto-request string' (unchecked). Below these are three text input fields: 'Username name attribute' with 'DomainUserName', 'Username ID attribute' with 'DomainUserName', and 'OTC Field' with 'otc'. At the bottom are 'Save' and 'Close' buttons. On the right side of the dialog, there is a vertical list of excluded URLs, which is partially obscured by a vertical yellow bar.

Show TURing image check to display the TURing image

Show Request String check to display a button to request the dual channel security string to send to the user

Show Pinpad check to display a Pinpad keypad

Show blank image for unknown user if checked, no image is shown if the user is not know. If unchecked, a random image is shown.

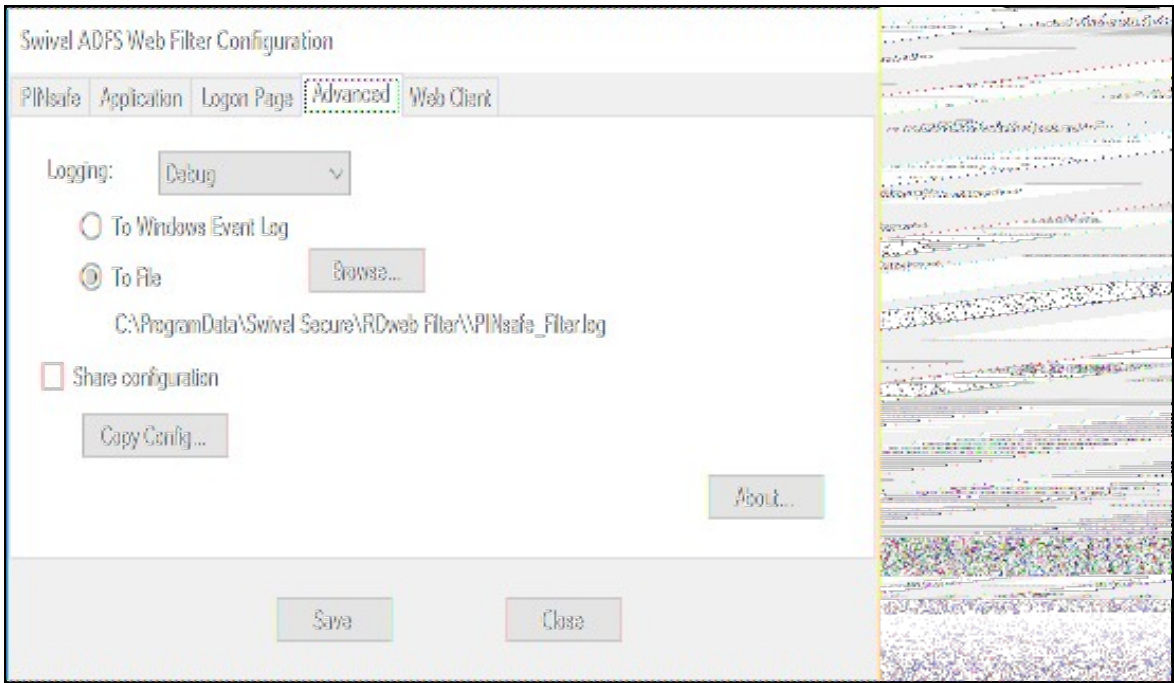
Auto-display image if checked, the TURing or Pinpad is automatically displayed after entering the username.

Auto-request string if checked, a security string is automatically requested after entering the username.

Username name attribute the HTML "name" attribute for the username field. Do not change this unless instructed.

Username ID attribute the HTML "id" attribute for the username field. Do not change this unless instructed.

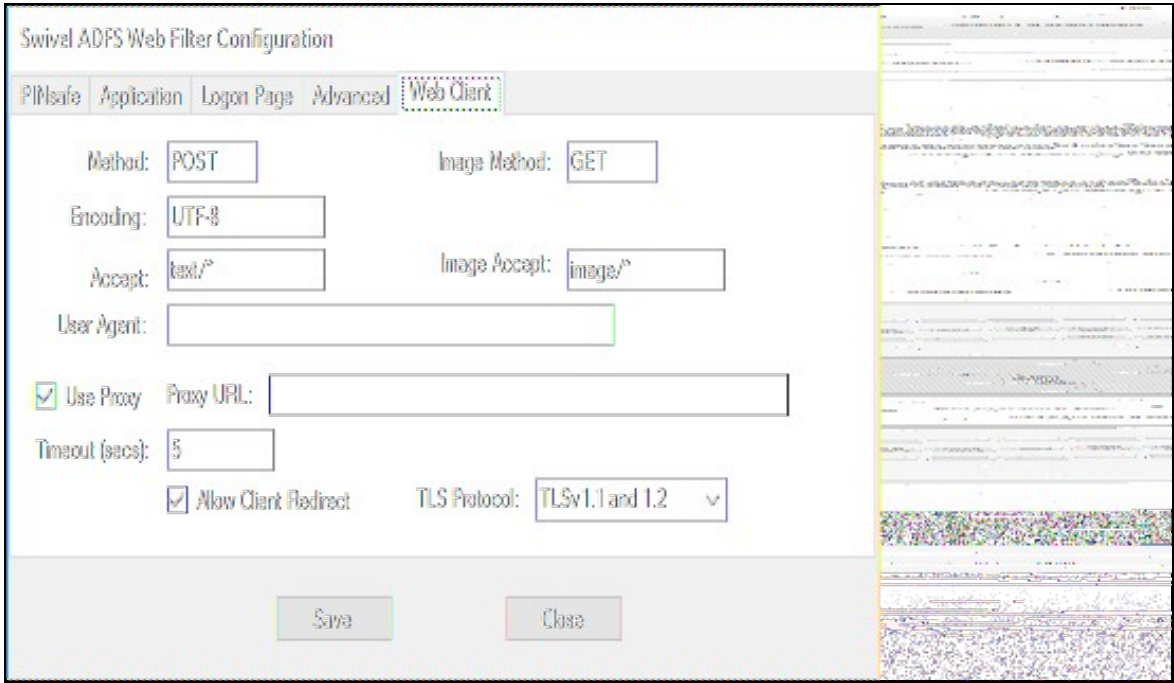
OTC Field the HTML "name" attribute for the OTC field. Do not change this unless instructed.



Logging enables the recording of certain information by the filter. The different levels indicate more detailed logs. Logs can either be written to the Windows Event Log, or to a chosen file. When writing to a file, make sure that the account used to run the RDWeb application has write access to the appropriate folder.

Share configuration allows you to export the configuration and import it to another RDWeb server.

About displays the version number and copyright information.



Most of the settings on this page should be left unchanged, unless instructed. The one exception is

TLS Protocol Version 2 Swivel appliances do not support TLS versions 1.1 or 1.2. Version 3 and 4 appliances do not support anything lower than TLS 1.1 unless specifically enabled, so unless you have a version 2 appliance, please ensure that you select "TLSv1.1 and 1.2".

If you need to change any of these settings later, a link to the configuration program is provided on the shortcut menu.

26 Changes to Existing Files

The installer will make modifications to three files within the RDS web hierarchy:

- Login.aspx from within the language folder. The appropriate buttons to display a TURing image are added if required. If you have significantly altered the login page, the installer may not be able to make its changes. Contact Swivel Secure for advice in this case.
- Renderscripts.js. A new function is added to display a TURing image, or to request a message on demand.
- Web.config. The Swivel filter is added as a new module, and the Swivel server details are stored under appSettings.

Additionally, the filter copies two DLLs to the bin folder of RDWeb/Pages: the filter itself and the Swivel client. It also copies a TURing image proxy, pinsafe_image.aspx, to the language folder.

27 Troubleshooting

We have seen in one instance, a problem whereby the TURING image could not be displayed even though the settings were correct, and the TURING image could be directly requested from the RDS Web server to the Swivel virtual or hardware appliance. The conclusion in this case was that the problem was due to permissions issues with the RDSWeb application pool account. Although we were unable to identify the exact problem, we resolved it by changing a setting on the application pool (under Advanced Settings) to enable Load User Profile.

28 Uninstalling

An uninstall program is provided, so you can either uninstall from the Windows Control Panel, or from the uninstall link on the shortcut menu.

The uninstall process requires that the files `login.aspx.sav` and `renderscripts.js.sav`, which are created when the appropriate files are modified, remain in their initial locations. These are the original files, without the PINsafe modifications. If these files do not exist, the filter cannot be properly uninstalled.

29 Microsoft Sharepoint 2010 Integration

29.1 Overview

The solution described here is for SharePoint 2010 only, as it relies on claims-based authentication features introduced in that version. A similar solution is also available for [SharePoint 2013](#).

For earlier versions of SharePoint, see [this article](#).

29.2 Downloading the Filter

The latest solution for PINsafe SharePoint 2010 filter is version 1.5.3. It can be found [here](#). Full instructions for installing the filter and configuring SharePoint to support it are included in the zip file.

29.3 Upgrading the Swivel SharePoint Filter

It is recommended to uninstall the old filter and then reinstall the new filter in order to remove the previous filter from the assembly cache. Choose the appropriate upgrade option when installing the new version.

29.4 Uninstalling the Swivel SharePoint filter

To uninstall the filter select the Add/Remove Programs on the SharePoint server then uninstall. The settings are not removed in the uninstall process.

29.5 SharePoint PINsafe FAQ

29.5.1 Can the Swivel SharePoint filter Manage the AD Credentials

Yes, version 1.4 or later of the SharePoint filter has the ability to set the AD password. See documentation included in the download for more information.

29.5.2 Can the filter detect an expired AD password

Yes, version 1.4 or later of the filter will detect if the AD password has expired, and will redirect to a page suitable for changing the AD password. See documentation included in the download for more information.

29.5.3 Can the filter allow users to reset a forgotten AD password

Yes, version 1.5 or later of the SharePoint filter has the ability to reset forgotten AD passwords as well as PINs. See included documentation for more information.

Note that the password reset feature requires version 3.9.6 or later of the Swivel Core server. However, if you do not wish to upgrade, a patch is available for version 3.8, from [here](#), to add the required feature. If you want to use this feature, please contact support@swivelsecure.com to check if your version of PINsafe can be upgraded to support this feature. Please also contact support@swivelsecure.com for help in installing this patch.

29.5.4 Can the filter work with Membership Providers other than Active Directory

Yes, version 1.5.2 or later will work with other membership providers. However, the AD password management facilities must be disabled. These only work with the AD membership provider, and authentication with other membership providers will fail if this feature is enabled.

29.6 Troubleshooting

29.6.1 TURing image does not appear

A red cross may be present where the TURing image should appear. The usual causes are:

Incorrect path to PINsafe appliance

Self Signed Certificate is used, but the allow self signed certificate option has not been selected

Firewall blocking access to PINsafe server

Network issue

HTTP request against the PINsafe running HTTPS

29.6.2 Error Messages

502 - Bad Gateway

This has been seen where the SharePoint cannot connect to the PINsafe server. Check the above, particularly the settings for SSL or HTTP access.

Authentication provider not found

PINsafe cannot be accessed by the SharePoint server, verify connectivity settings.

30 Microsoft Sharepoint 2013 Integration

30.1 Overview

The solution described here is for SharePoint 2013 only. A similar solution is available for [SharePoint 2010](#). Do not use version 1.6 of the filter for SharePoint 2010, and do not use earlier versions for SharePoint 2013.

For earlier versions of SharePoint, see [this article](#).

30.2 Downloading the Filter

The latest solution for PINsafe SharePoint 2013 filter is version 1.6.1. It can be found [here](#). The only change from 1.6.0 is that removing the domain prefix and/or suffix from usernames is now optional. In 1.6.0, they were always removed.

Full instructions for installing the filter and configuring SharePoint to support it are included in the zip file, or you can download it separately from [here](#).

The previous version, 1.6.0, can be found [here](#).

30.3 Upgrading the Swivel SharePoint Filter

It is recommended to uninstall the old filter and then reinstall the new filter in order to remove the previous filter from the assembly cache. You should then choose the appropriate upgrade option when prompted. Note that upgrading from a SharePoint 2010 filter has not been tested, but as the technology is very similar, no problems are anticipated. If you do have a problem when upgrading an existing SharePoint 2010 filter to SharePoint 2013, it is recommended that you uninstall and treat it as a new installation.

30.4 Uninstalling the Swivel SharePoint filter

To uninstall the filter select the Add/Remove Programs on the SharePoint server then uninstall. The settings are not removed in the uninstall process.

30.5 SharePoint PINsafe FAQ

30.5.1 Can the Swivel SharePoint filter Manage the AD Credentials

Yes, version 1.4 or later of the SharePoint filter has the ability to set the AD password. See documentation included in the download for more information.

30.5.2 Can the filter detect an expired AD password

Yes, version 1.4 or later of the filter will detect if the AD password has expired, and will redirect to a page suitable for changing the AD password. See documentation included in the download for more information.

30.5.3 Can the filter allow users to reset a forgotten AD password

Yes, version 1.5 or later of the SharePoint filter has the ability to reset forgotten AD passwords as well as PINs. See included documentation for more information.

However, the password reset feature requires version 3.9.6 or later of the Swivel Core server, which at the time of writing had not been released. However, a patch is available for version 3.8, from [here](#), to add the required feature. If you want to use this feature, please contact support@swivelsecure.com to check if your version of PINsafe can be upgraded to support this feature. Please also contact support@swivelsecure.com for help in installing this patch.

30.5.4 Can the filter work with Membership Providers other than Active Directory

Yes, version 1.5.2 or later will work with other membership providers. However, the AD password management facilities must be disabled. These only work with the AD membership provider, and authentication with other membership providers will fail if this feature is enabled.

30.6 Troubleshooting

30.6.1 TURING image does not appear

A red cross may be present where the TURING image should appear. The usual causes are:

Incorrect path to PINsafe appliance

Self Signed Certificate is used, but the allow self signed certificate option has not been selected

Firewall blocking access to PINsafe server

Network issue

HTTP request against the PINsafe running HTTPS

30.6.2 Error Messages

502 - Bad Gateway

This has been seen where the SharePoint cannot connect to the PINsafe server. Check the above, particularly the settings for SSL or HTTP access.

Authentication provider not found

PINsafe cannot be accessed by the SharePoint server, verify connectivity settings.

31 Microsoft Sharepoint 2019 Integration

31.1 Overview

NOTE: this solution is based on the SharePoint 2013 solution. As such, it has had limited testing on SharePoint 2019, but it appears to be working successfully.

The solution described here is for SharePoint 2019 only. Similar solutions are available for [SharePoint 2013](#) and [SharePoint 2010](#). Do not use version 1.8 of the filter for previous versions of SharePoint, and do not use earlier versions for SharePoint 2019.

Please note that the illustrations in this article are from the SharePoint 2013 integration. The forms will look slightly different in 2019, but functionality is essentially the same. There may also be some outdated references to SharePoint 2013. This article will be updated in due course.

31.2 Downloading the Filter

The latest solution for PINsafe SharePoint 2019 filter is version 1.8.0. It can be found [here](#).

Full instructions for installing the filter and configuring SharePoint to support it can be downloaded from [here](#). This article refers to version 1.6 for SharePoint 2013, but the instructions are unchanged.

31.3 Upgrading the Swivel SharePoint Filter

It is recommended to uninstall the old filter and then reinstall the new filter in order to remove the previous filter from the assembly cache. You should then choose the appropriate upgrade option when prompted. Note that upgrading from a SharePoint 2010 filter has not been tested, but as the technology is very similar, no problems are anticipated. If you do have a problem when upgrading an existing SharePoint 2010 filter to SharePoint 2013, it is recommended that you uninstall and treat it as a new installation.

31.4 Uninstalling the Swivel SharePoint filter

To uninstall the filter select the Add/Remove Programs on the SharePoint server then uninstall. The settings are not removed in the uninstall process.

31.5 SharePoint PINsafe FAQ

31.5.1 Can the Swivel SharePoint filter Manage the AD Credentials

Yes, version 1.4 or later of the SharePoint filter has the ability to set the AD password. See documentation included in the download for more information.

31.5.2 Can the filter detect an expired AD password

Yes, version 1.4 or later of the filter will detect if the AD password has expired, and will redirect to a page suitable for changing the AD password. See documentation included in the download for more information.

31.5.3 Can the filter allow users to reset a forgotten AD password

Yes, version 1.5 or later of the SharePoint filter has the ability to reset forgotten AD passwords as well as PINs. See included documentation for more information.

However, the password reset feature requires version 3.9.6 or later of the Swivel Core server, which at the time of writing had not been released. However, a patch is available for version 3.8, from [here](#), to add the required feature. If you want to use this feature, please contact support@swivelsecure.com to check if your version of PINsafe can be upgraded to support this feature. Please also contact support@swivelsecure.com for help in installing this patch.

31.5.4 Can the filter work with Membership Providers other than Active Directory

Yes, version 1.5.2 or later will work with other membership providers. However, the AD password management facilities must be disabled. These only work with the AD membership provider, and authentication with other membership providers will fail if this feature is enabled.

31.6 Troubleshooting

31.6.1 TURING image does not appear

A red cross may be present where the TURING image should appear. The usual causes are:

Incorrect path to PINsafe appliance

Self Signed Certificate is used, but the allow self signed certificate option has not been selected

Firewall blocking access to PINsafe server

Network issue

HTTP request against the PINsafe running HTTPS

31.6.2 Error Messages

502 - Bad Gateway

This has been seen where the SharePoint cannot connect to the PINsafe server. Check the above, particularly the settings for SSL or HTTP access.

Authentication provider not found

PINsafe cannot be accessed by the SharePoint server, verify connectivity settings.

32 Microsoft Sharepoint Integration Methods

32.1 Overview

This document describes methods of integrating Swivel authentication with earlier versions of SharePoint that do not support claims-based authentication. Although these methods will work with later versions of SharePoint, it is recommended that you use the appropriate dedicated filters for SharePoint 2010 and 2013, in the following links:

[SharePoint 2010](#)

[SharePoint 2013](#)

32.2 Integration Using TMG or ISA

Our recommended solution for earlier versions of SharePoint is to use Microsoft TMG integration with RADIUS authentication (see [here](#) or [here](#)), or Microsoft ISA Server with RADIUS authentication (see [Microsoft ISA 2006 Integration](#)). However, the following article shows how to integrate with SharePoint as a 2-stage authentication process.

32.3 Authenticating to Earlier Versions of SharePoint as a 2-Stage Process

The solution is to use the [PINsafe IIS7 filter](#). Install as per the included instructions.

The result should be that you will need to authenticate first to the Active Directory domain, if you are not already logged in. Subsequently, you will be redirected to the PINsafe login page to complete the second part of the authentication process, before being finally redirected to the SharePoint home page.

One issue which is not addressed by the IIS filter documentation, which might cause problems, particularly in Windows 2008 Server, is that the Windows account running the SharePoint application (normally Network Service) needs to have read and execute permission on the pinsafe virtual directory.

33 Open ERP7 Integration

34 Overview

This document outlines the steps required to integrate the OpenERP 7.0 with Swivel using dual or single channel authentication.

The Swivel install requires configuring an agent on the Swivel server and setting up a shared secret with the code being added to OpenERP 7.0 server to allow communication for authentication.

Due to the complexity of OpenERP, several files need to be changed (or replaced) on the server, then recompiled. Integration with PINsafe is made through Agent-XML.

NOTE: This document refers to the version 7.0 of OpenERP. If using version 7.0 you can replace the files with a simple copy paste, else please refer to the KB article about [OpenERP_Custom_Integration](#).

35 Prerequisites

OpenERP 7.0 server

Swivel server

The files that need to be replaced: [Media:OpenERP7.tar.gz](#)

36 Swivel Configuration

On the Swivel server configure the agent that is permitted to request authentication. On the PINsafe Administration Console select from the server menu Agents and enter the details of the OpenERP IP address and a shared key, then click on apply. Example:

Name : OpenERP,
Hostname/IP : 10.10.10.252,
shared secret : secret

The screenshot shows the Swivel Administration Console interface. At the top left is the Swivel logo with the tagline "the power of knowing". A navigation menu on the left lists various configuration options, with "Agents" selected under the "Server" category. The main content area is titled "Server > Agents" and contains a list of agents, with "local" selected. Below the list is a configuration form for the selected agent. The form fields are: Name (OpenERP), Can act as Repository (Yes), Hostname/IP (10.10.10.252), Shared secret (masked with dots), Group (---ANY---), Authentication Modes (ALL), Check password with Repository (No), Username attribute for repository (empty), Allow alternative usernames (No), and Alternative username attributes (empty). There are "Apply" and "Delete" buttons at the bottom right of the form area.

If Single Channel communication is to be used, select from the PINsafe Administration Console Single Channel, and set the Allow image request by username to Yes then click on apply.

- [Status](#)
- [Log Viewer](#)
- [-] [Server](#)
 - ◊ [Name](#)
 - ◊ [Language](#)
 - ◊ [License](#)
 - ◊ [Jobs](#)
 - ◊ [SMTP](#)
 - ◊ [Agents](#)
 - ◊ [Peers](#)
 - ◊ [Single Channel](#)
 - ◊ [Dual Channel](#)
 - ◊ [Third Party Authentication](#)
 - ◊ [Voice Channel](#)
- [-] [Policy](#)
- [-] [Logging](#)
- [-] [Transport](#)
- [-] [Database](#)
- [-] [Mode](#)
- [-] [Repository](#)
- [-] [RADIUS](#)
- [-] [Migration](#)
- [-] [Appliance](#)
- [-] [OATH](#)
- [-] [Synchronisation Administration](#)
- [-] [Reporting](#)
- [User Administration](#)
- [Save Configuration](#)
- [Administration Guide](#)
- [Logout](#)

Server > Single Channel

Please specify how single channel security strings are delivered.

Allow session request by username:	<input style="width: 40px; height: 20px; border: 1px solid #ccc;" type="button" value="Yes"/>
Allow alternative usernames:	<input style="width: 40px; height: 20px; border: 1px solid #ccc;" type="button" value="No"/>
Alternative username attributes:	<input style="width: 150px; height: 20px; border: 1px solid #ccc;" type="text"/>
Multiple Authentications per String:	<input style="width: 40px; height: 20px; border: 1px solid #ccc;" type="button" value="No"/>
Image file:	<input style="width: 100px; height: 20px; border: 1px solid #ccc;" type="text" value="turingOrange.xml"/>
Background image file:	<input style="width: 150px; height: 20px; border: 1px solid #ccc;" type="text" value="backgroundsMonochrome.xml"/>
Text Alpha Value:	<input style="width: 60px; height: 20px; border: 1px solid #ccc;" type="text" value="100"/>
Only use one font per image:	<input style="width: 40px; height: 20px; border: 1px solid #ccc;" type="button" value="Yes"/>
Image Rendering:	<input style="width: 40px; height: 20px; border: 1px solid #ccc;" type="text" value="Static"/>
Jiggle characters within slot:	<input style="width: 40px; height: 20px; border: 1px solid #ccc;" type="button" value="No"/>
Add blank trailer frame to animated images:	<input style="width: 40px; height: 20px; border: 1px solid #ccc;" type="button" value="No"/>
Number of complete display cycles per image:	<input style="width: 60px; height: 20px; border: 1px solid #ccc;" type="text" value="10"/>
Inter-frame delay (1/100s):	<input style="width: 60px; height: 20px; border: 1px solid #ccc;" type="text" value="25"/>
No. Characters Visible:	<input style="width: 60px; height: 20px; border: 1px solid #ccc;" type="text" value="1"/>

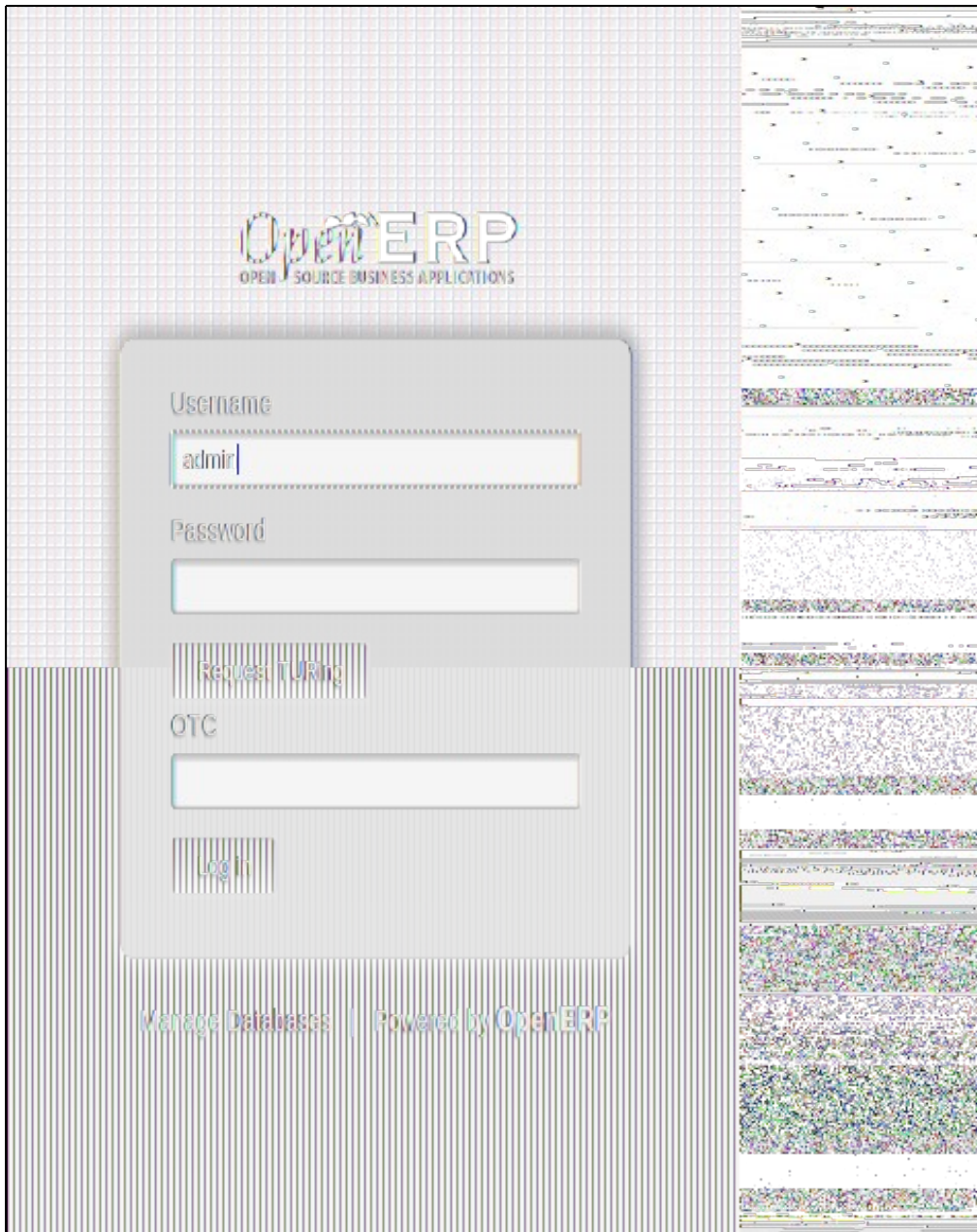
37 Configuring the OpenERP 7 server

37.1 Upload the custom files to OpenERP 7

1. Shutdown OpenERP 7 server service.
2. Unpack the "File Package" (you can find it in the Prerequisites section of this page) to a local directory.
3. Edit the file OpenERP/addons/web/controllers/main.py Search for ?10.10.10.201? (line 859) and change it with the IP address or domain name for your Swivel server
4. Edit the file OpenERP/addons/web/controllers/main.py Search for ?<Secret>secret<%2FSecret>? (line 860) and change it with ?<Secret>THE SECERT YOU CHOSE IN THE AGENT CONFIG<%2FSecret>?
5. Edit the file OpenERP/addons/web/static/source/js/chrome.js Search for ?10.10.10.201? (line 710) and change it with the IP address or domain name for your Swivel server
6. FTP ALL the files to the server running OpenERP, respecting the directory structure and overwriting
7. Start OpenERP Server with (recompile option enabled -u) : /?path to openERP?/openerp-server -u ?database to run?

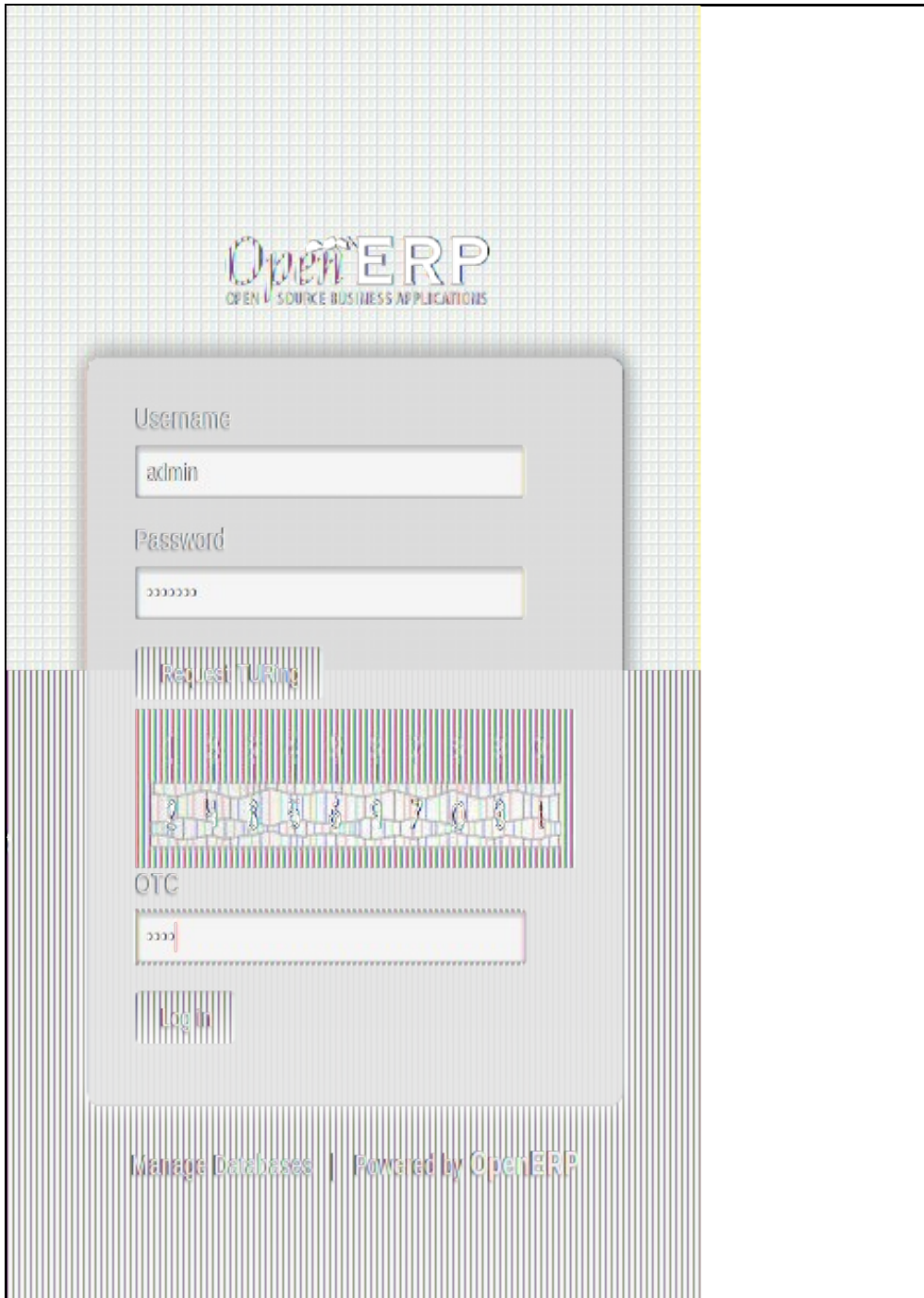
37.2 Testing

Open a browser and point to <http://yourserverip:8069>



After clicking the ?request TURing? button, the swivel appliance log should show ?127.0.0.1:Session started for user: admin.?

And this should be the result:



Then after login, the swivel appliance log should read:

?OPENERP SERVER IP? OpenERP:Login successful for user: admin.

38 Error Messages

38.1 On OpenERP stack trace

38.1.1 "ImportError: No module named urllib.urlopen"

Please refer to the [OpenERP_Custom_Integration](#) article, section Python 3.0 changes.

38.2 On Swivel Log

38.2.1 AgentXML request failed, error: The agent is not authorized to access the server

User fails to authenticate with the above error message in the Swivel log. An Agent on Swivel server has not been defined for the OpenERP server. Go to Server/Agents in the Swivel admin console, and add a new entry, using the IP address of the OpenERP server. Make sure the agent secret is the same as on the OpenERP configuration.

39 OpenERP Custom Integration

40 Overview

This document outlines the steps required to integrate the OpenERP with Swivel using dual or single channel authentication.

The Swivel install requires configuring an agent on the Swivel server and setting up a shared secret with the code being added to OpenERP 7.0 server to allow communication for authentication.

Due to the complexity of OpenERP, several files need to be changed (or replaced) on the server, then recompiled. Integration with Swivel is made through Agent-XML.

NOTE: This document uses the files from version 7.0 of OpenERP, however, the principle is that either customized, or from another version, the functions should be similar, making integration easier.

For an article specific for version OpenERP 7.0, please check article [Open_ERP7_Integration](#)

41 Prerequisites

OpenERP server (version 7.0 assumed in this example)

PINsafe server

Text Editor or Code editor

It is recommended to read the following Knowledgebase articles (to better understand how Swivel Agent-XM works):

<https://kb.swivelsecure.com/wiki/index.php/Agent-XML>

<https://kb.swivelsecure.com/wiki/index.php/AuthenticationAPI>

42 Swivel Configuration

On the Swivel server configure the agent that is permitted to request authentication. On the Swivel Administration Console select from the server menu Agents and enter the details of the OpenERP IP address and a shared key, then click on apply. Example:

Name : OpenERP,
Hostname/IP : 10.10.10.252,
shared secret : secret

The screenshot shows the Swivel Administration Console interface. At the top left is the Swivel logo with the tagline "the power of knowing". A navigation menu on the left lists various system settings, with "Agents" selected under the "Server" category. The main content area is titled "Server > Agents" and contains a list of agents, with "local" selected. Below the list is a configuration form for the selected agent. The form fields are: Name (OpenERP), Can act as Repository (Yes), Hostname/IP (10.10.10.252), Shared secret (masked with dots), Group (---ANY---), Authentication Modes (ALL), Check password with Repository (No), Username attribute for repository (empty), Allow alternative usernames (No), and Alternative username attributes (empty). There are "Apply" and "Delete" buttons at the bottom right of the form.

If Single Channel communication is to be used, select from the Swivel Administration Console Single Channel, and set the Allow image request by username to Yes then click on apply.

- [Status](#)
- [Log Viewer](#)
- [-] [Server](#)
 - ◊ [Name](#)
 - ◊ [Language](#)
 - ◊ [License](#)
 - ◊ [Jobs](#)
 - ◊ [SMTP](#)
 - ◊ [Agents](#)
 - ◊ [Peers](#)
 - ◊ [Single Channel](#)
 - ◊ [Dual Channel](#)
 - ◊ [Third Party Authentication](#)
 - ◊ [Voice Channel](#)
- [-] [Policy](#)
- [-] [Logging](#)
- [-] [Transport](#)
- [-] [Database](#)
- [-] [Mode](#)
- [-] [Repository](#)
- [-] [RADIUS](#)
- [-] [Migration](#)
- [-] [Appliance](#)
- [-] [OATH](#)
- [-] [Synchronisation Administration](#)
- [-] [Reporting](#)
- [User Administration](#)
- [Save Configuration](#)
- [Administration Guide](#)
- [Logout](#)

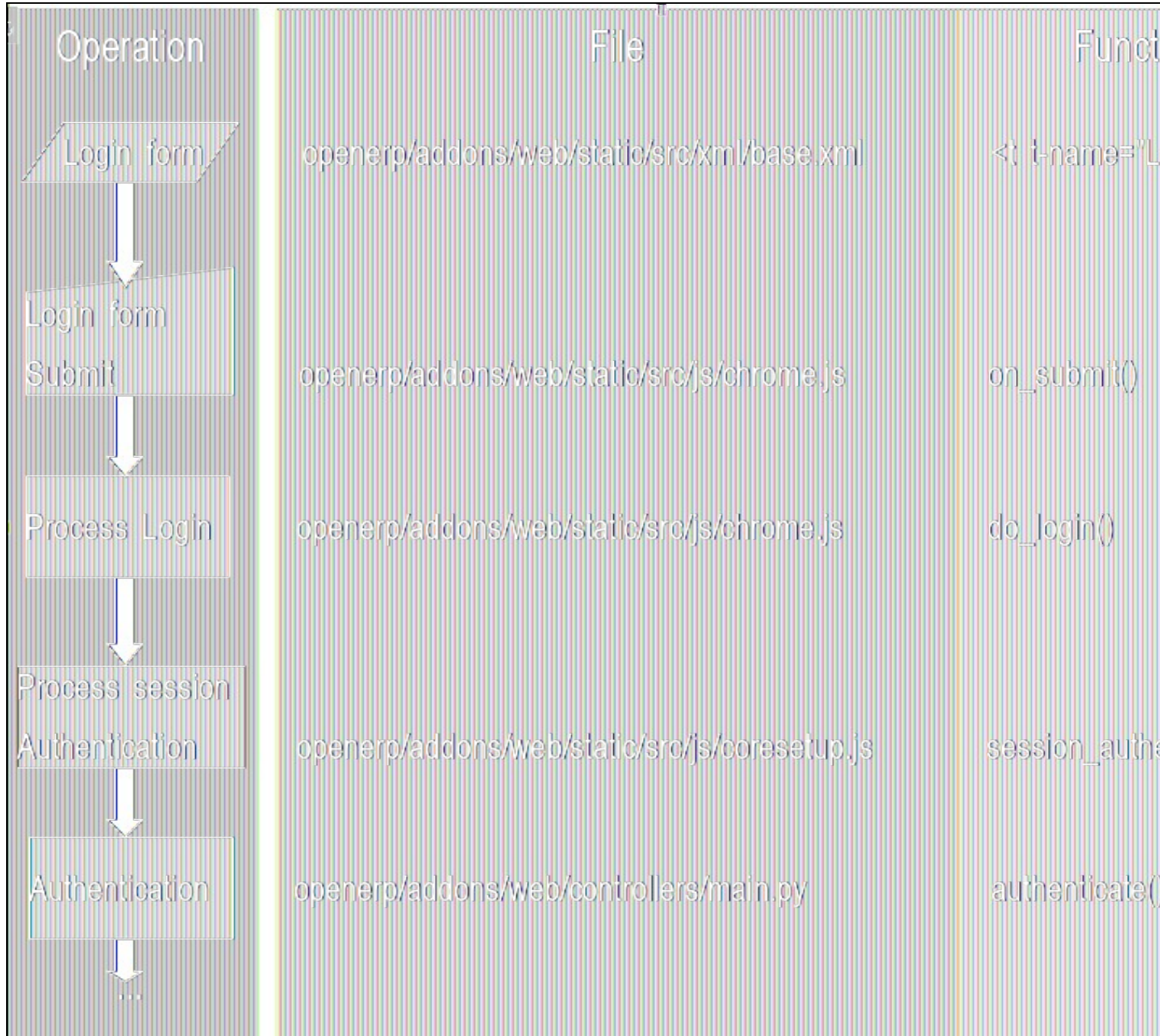
Server > Single Channel

Please specify how single channel security strings are delivered.

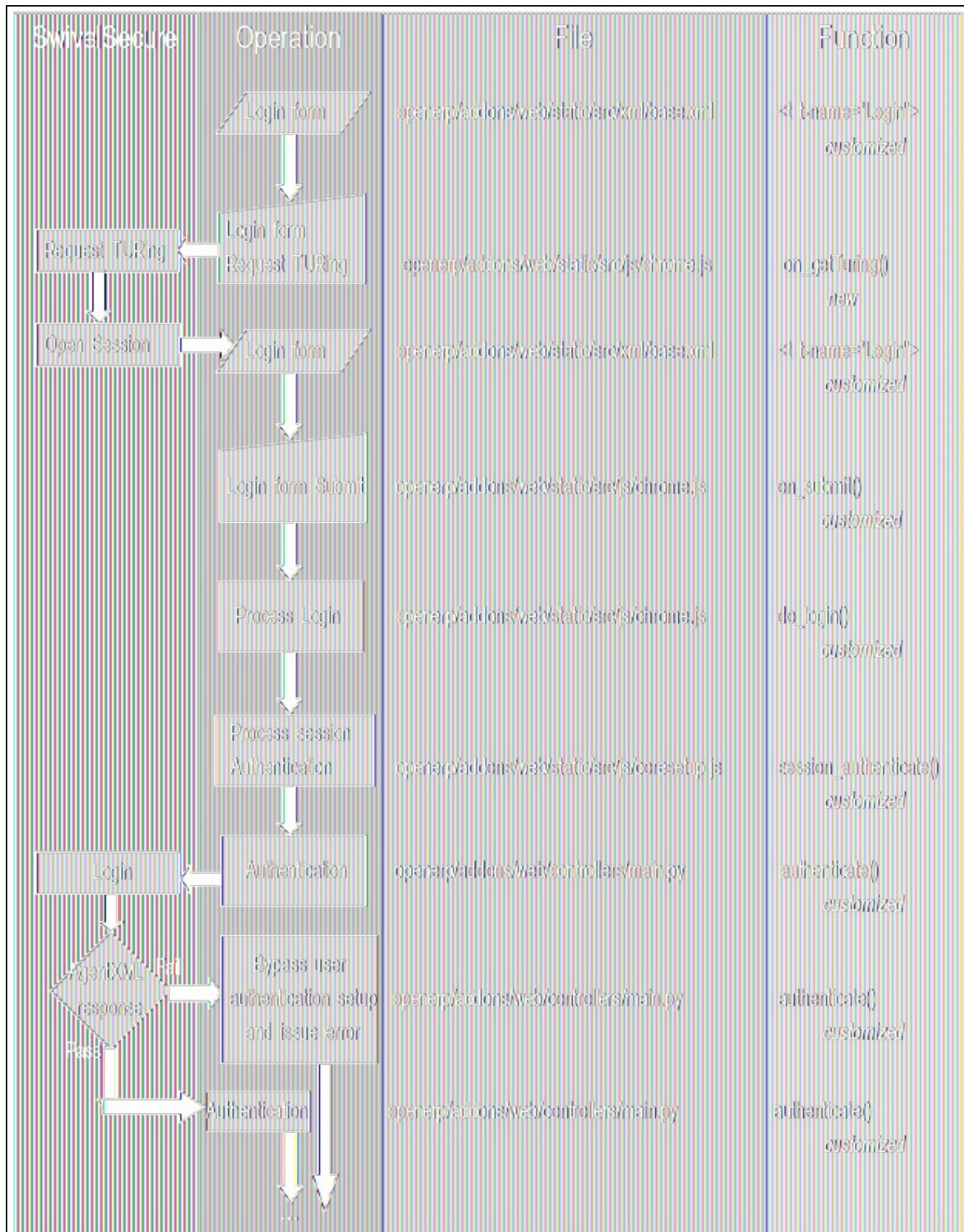
Allow session request by username:	<input style="width: 40px; height: 20px; border: 1px solid #ccc;" type="button" value="Yes"/>
Allow alternative usernames:	<input style="width: 40px; height: 20px; border: 1px solid #ccc;" type="button" value="No"/>
Alternative username attributes:	<input style="width: 150px; height: 20px; border: 1px solid #ccc;" type="text"/>
Multiple Authentications per String:	<input style="width: 40px; height: 20px; border: 1px solid #ccc;" type="button" value="No"/>
Image file:	<input style="width: 100px; height: 20px; border: 1px solid #ccc;" type="text" value="turingOrange.xml"/>
Background image file:	<input style="width: 150px; height: 20px; border: 1px solid #ccc;" type="text" value="backgroundsMonochrome.xml"/>
Text Alpha Value:	<input style="width: 60px; height: 20px; border: 1px solid #ccc;" type="text" value="100"/>
Only use one font per image:	<input style="width: 40px; height: 20px; border: 1px solid #ccc;" type="button" value="Yes"/>
Image Rendering:	<input style="width: 40px; height: 20px; border: 1px solid #ccc;" type="text" value="Static"/>
Jiggle characters within slot:	<input style="width: 40px; height: 20px; border: 1px solid #ccc;" type="button" value="No"/>
Add blank trailer frame to animated images:	<input style="width: 40px; height: 20px; border: 1px solid #ccc;" type="button" value="No"/>
Number of complete display cycles per image:	<input style="width: 60px; height: 20px; border: 1px solid #ccc;" type="text" value="10"/>
Inter-frame delay (1/100s):	<input style="width: 60px; height: 20px; border: 1px solid #ccc;" type="text" value="25"/>
No. Characters Visible:	<input style="width: 60px; height: 20px; border: 1px solid #ccc;" type="text" value="1"/>

43 OpenERP User Authentication flow Integration

43.1 The Standard OpenERP User Authentication flow



43.2 The Swivel Integrated OpenERP User Authentication flow



44 Configuring the OpenERP Files

44.1 openerp/addons/web/static/src/xml/base.xml

Unchanged code in black

Added code in green

Changed code in orange

In this file we are transforming the form to support the Get TURING button and the OTC password field.

Search for the login form code, usually called `<t t-name="Login">?` and place in line nr 61

```
Code
<t t-name="Login">
  <div class="oe_login">
    <div class="oe_login_bottom"> </div>
    <div class="oe_login_error_message"/>
    <div class="oe_login_pane">
      <div class="oe_login_logo"></div>
      <form action="" method="post">
        <div class="oe_login_dbpane" >
          Database:
          <input name="db" type="text" value="" />
        </div>
        <ul>
          <li>Username</li>
          <li><input name="login" type="text" value="" autofocus="autofocus" /></li>
          <li>Password</li>
          <li><input name="password" type="password" value="" /></li>
          <li><button name="getTURING">Request TURING</button></li>
          <div class="TURING"><img name="TURINGimg" class="myTURINGimg" /></div>
          <li>OTC</li>
          <li><input name="password2" type="password" value="" /></li>
          <li><button name="submit">Log in</button></li>
        </ul>
      </form>
      <div class="oe_login_footer">
        <a href="#" class="oe_login_manage_db">Manage Databases</a> |
        <a href="http://www.openerp.com" target="_blank">Powered by <span>OpenERP</span></a>
      </div>
    </div>
  </div>
</t>
```

44.2 openerp/addons/web/static/src/js/chrome.js

Unchanged code in black

Added code in green

Changed code in orange

In this file we will be adding the functions that support automation and control over the form displaying the TURING image and Get TURING button.

First we need to hide the TURING image and register the link from the Get TURING button to the respective function.

Search for the form initiation routine, usually called ? start: function() {? and place in line nr 640

Code

```
start: function() {
    var self = this;
    self.$el.find("form").submit(self.on_submit);
    this.$(dm.TURING).hide();
    self.$el.find("form button[name=getTURING]").click(self.on_getTuring);
    self.$el.find("#oe_login_manage_db").click(function() {
        self.do_action("database_manager");
    });
    self.on("change:database_selector", this, function() {
        this.database_selected(this.get("database_selector"));
    });
    var d = $.when();
    if ($.param.fragment().token) {
        self.params.token = $.param.fragment().token;
    }
    // used by dbmanager.do_create via internal client action
    if (self.params.db && self.params.login && self.params.password) {
        d = self.do_login(self.params.db, self.params.login, self.params.password);
    } else {
        d = self.rpc("/web/database/get_list", {})
            .done(self.on_db_loaded)
            .fail(self.on_db_failed)
            .always(function() {
                if (self.selected_db && self.has_local_storage && self.remember_credentials) {
                    self.$("[name=login]").val(localStorage.getItem(self.selected_db + "last_login") || "");
                }
            });
    }
    return d;
},
```

Then we need to create the function. Find space right after the ?on_db_failed: function? on lines 703 to 707.

Change the IP address on the example with your swivel appliance IP or domain name.

Code

```
on_do_failed: function (error, event) {
    if (error.data.fault_code === 'AccessDenied') {
        event.preventDefault();
    }
},
on_getTuring: function() {
    var login = this.$("form input[name='login']").val();
    var SwivelURL = 'https://10.10.10.201:8443/proxy/SCImage?username=' + login + "&random=" + Math.floor(Math.random() * 10000);
    this.$("myTURINGing").attr('src', SwivelURL);
    this.$("div.TURING").show();
    return false;
},
```

Finally, we need to change the functions that support the original form submit, so we can add OTC (called "password2" in this example).

Function ?on_submit: function(ev)? should be immediately after your recently added function code, on line nr 715.

You will also update function ?do_login?, placed immediately after ?on_submit? function.

Code

```
on_submit: function(ev) {
    if(ev) {
        ev.preventDefault();
    }
    var db = this.$("form [name=db]").val();
    if (!db) {
        this.do_warn(_("Login"), _("No database selected !"));
        return false;
    }
    var login = this.$("form input[name=login]").val();
    var password = this.$("form input[name=password]").val();
    var password2 = this.$("form input[name=password2]").val();
    this.do_login(db, login, password, password2);
},
/*
 * Performs actual login operation, and UI-related stuff
 *
 * @param {String} db database to log in
 * @param {String} login user login
 * @param {String} password user password
 */
do_login: function (db, login, password, password2) {
    var self = this;
    self.hide_error();
    self.$("#oe_login-pane").fadeOut("slow");

    return this.session.session_authenticate(db, login, password, password2).then(function() {
        self.remember_last_used_database(db);
        if (self.has_local_storage && self.remember_credentials) {
            localStorage.setItem(db + 'last_login', login);
        }
        self.trigger("login_successful");
    }, function () {
        self.$("#oe_login-pane").fadeIn("fast", function() {
            self.show_error(_("Invalid username or password"));
        });
    });
},
},
```

44.3 openerp/addons/web/static/src/js/coresetup.js

Unchanged code in black

Added code in green

Changed code in orange

In this file we will be adding the parameter `?password2?` to the function that creates the bridge between the user interface (in XML, javascript and jquery) and the OpenERP server core (in python).

Search for the `?session_authenticate: function?`, usually place on line nr101

Code

```
session.authenticate: function(db, login, password, password2, _volatile) {
var self = this;
var base_location = document.location.protocol + '//' + document.location.host;
var params = { to: db, login: login, password: password, password2: password2, base_location: base_location };
return this.rpc('web/session/authenticate', params).then(function(result) {
    if (!result.uid) {
        return $.Deferred().reject();
    }
    _extend(self, result);
    if (!_volatile) {
        self.set_cookie('session_id', self.session_id);
    }
    return self.load_modules();
});
},
```

44.4 openerp/addons/web/controllers/main.py

Unchanged code in black

Added code in green

Changed code in orange

In this file we will be changing the OpenERP core authentication function, so that it calls the Swivel Secure server with a login function request, and then continues on the consideration of the reply given by swivel software.

First we need to add a library module to handle and parse the XML sent by the Swivel Agent-XML.

This is added in the very beginning of the code file.

Code

```
#!/usr/bin/perl -w
# coding: utf-8

import ast
import base64
import csv
import glob
import itertools
import logging
import operator
import datetime
import hashlib
import os
import re
import simplejson
import time
import urllib
import urllib2
import urlparse
import xmlrpclib
import zlib

from xml.etree import ElementTree
from StringIO import StringIO

from xml.dom.minidom import parseString

import babel.messages.pofile
import werkzeug.utils
import werkzeug.wrappers

try:
    import xbt
except ImportError:
    xbt = None

import openerp
import openerp.modules.registry
from openerp.tools.translate import _
from openerp.tools import config

from .. import http
openerpweb = http
```

Finally, we need to change the `def authenticate?` function to handle the the new `password2?` parameter, call the Swivel Server, parse the response and make decision based on the response.

The `def authenticate?` function should start in line nr 857 or 858

Don't forget to change the server IP in the code by your own Swivel server IP or domain path and the "secret".

Code

```
@ogonarpweb.jsonrequest
def authenticate(self, req, db, login, password, password2, base_location=None):

    url = 'https://10.10.10.201:8080/pinsafe/AgentXML'
    params = '%3Fxml-version%3D1.0' + encoding%3D'UTF-8'%3F<SASRequest><Version>3.6<%2FVersion><Secret>secret<%2FSecret>
    <Action>login<%2FAction><Username>' + login + '<%2FUsername><Password><%2FPassword><OTC>' + password2 + '<%2FOTC><%2FSASRequest>'
    data = urllib.urlopen(url + '?' + params).read()
    dom = parseString(data)
    xmlTag = dom.getElementsByTagName('Result')[0].toxml()
    xmlData = xmlTag.replace('<Result>', '').replace('</Result>', '')
    if xmlData == 'PASS':
        wsgienv = req.hitrequest.environ
        env = dict(
            base_location=base_location,
            HTTP_HOST=wsgienv['HTTP_HOST'],
            REMOTE_ADDR=wsgienv['REMOTE_ADDR'],
        )
        req.session.authenticate(db, login, password, env)
        return self.session_info(req)
    if xmlData == 'FAIL':
        return {'error': _('Error, Invalid credentials !'), 'title': _('No access')}
```

45 Python 3.x changes

45.1 openerp/addons/web/controllers/main.py

Unchanged code in black

Added code in green

Changed code in orange

If running Python 3.0, then you have a call to the urllib module.

The urllib <<http://docs.python.org/2/library/urllib.html#module-urllib>> module has been split into parts and renamed in Python 3 to urllib.request, urllib.parse, and urllib.error.

This is shown in RED in the example from the changes to be done on file ?openerp/addons/web/controllers/main.py?

Do not forget to change the IP an "secret" in the code by your own Swivel server IP or domain path.

```
Code

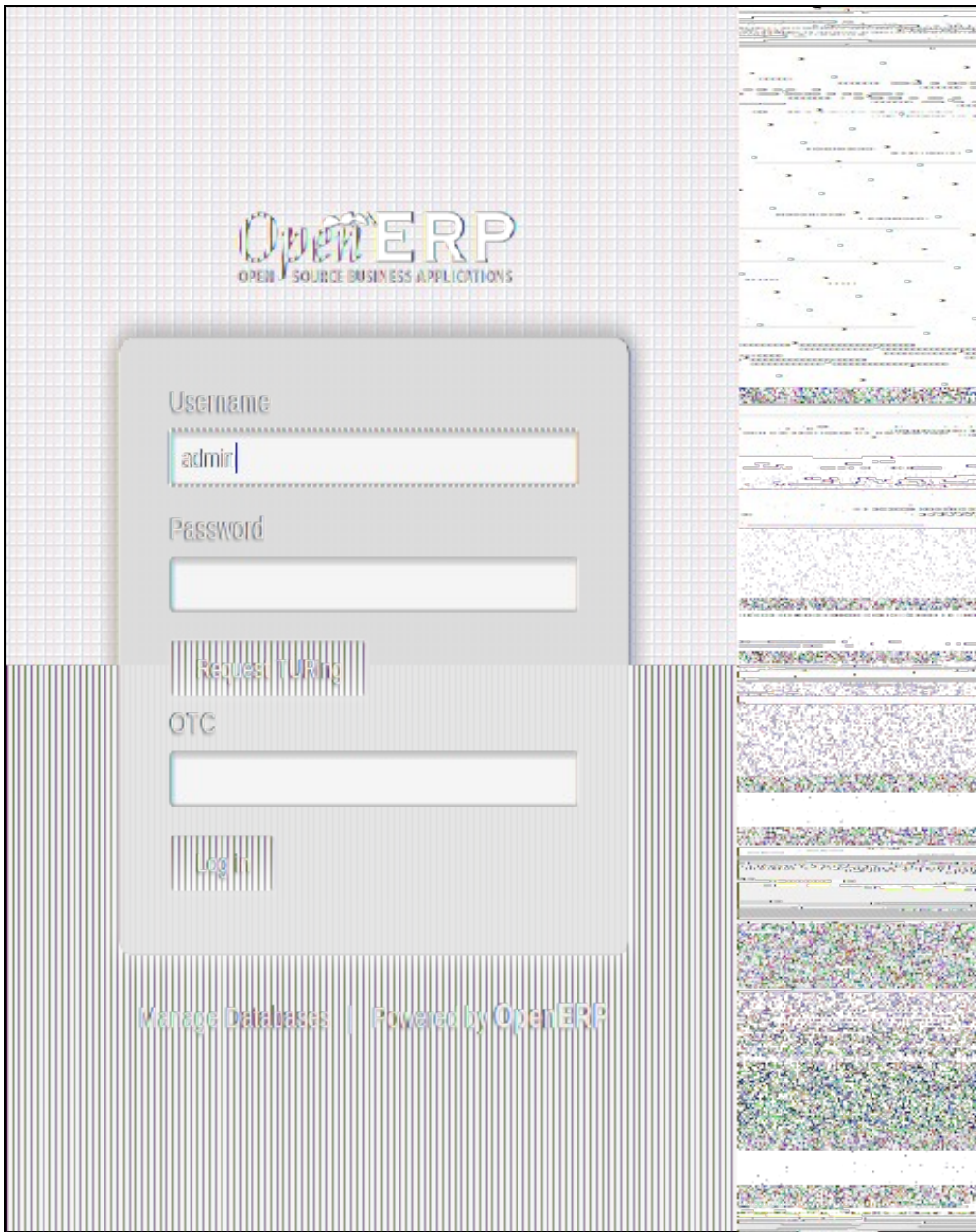
@openerpweb.jsonrequest
def authenticate(self, req, db, login, password, password2, base_location=None):

    url = 'https://10.10.10.201:8060/pinsefa/AgentXML'
    params = '<%3Fxml+version%3D'1.0'+encoding%3D'UTF-8'%2F><SASRequest><Version>3.6<%2FVersion><Secret>secret<%2FSecret>
    <Action>login<%2FAction><Username>' + login + '<%2FUsername><Password><%2FPassword><OTC>' + password2 + '<%2FOTC><%2FBASRequest>'

    data = urllib.request.urlopen(url + '?' + params).read()
    dom = parseString(data)
    xmlTag = dom.getElementsByTagName('Result')[0].toxml()
    xmlData = xmlTag.replace('<Result>', '').replace('</Result>', '')
    if xmlData == 'PASS':
        wsgienv = req.httprequest.environ
        env = dict(
            base_location=base_location,
            HTTP_HOST=wsgienv['HTTP_HOST'],
            REMOTE_ADDR=wsgienv['REMOTE_ADDR'],
        )
        req.session.authenticate(db, login, password, env)
        return self.session_info(req)
    if xmlData == 'FAIL':
        return {'error': _('Error, Invalid credentials !'), 'title': _('No access')}
```

45.2 Testing

Open a browser and point to <http://yourserverip:8069>



After clicking the ?request TURING? button, the swivel appliance log should show ?127.0.0.1:Session started for user: admin.?

And this should be the result:

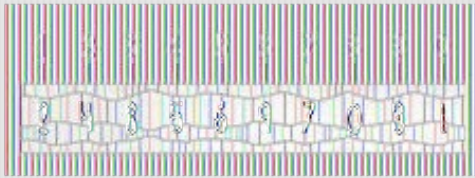


Username

admin

Password

Request Turing



OTC

Log In

Manage Databases | Powered by OpenERP

Then after login, the swivel appliance log should read:

?OPENERP SERVER IP? OpenERP:Login successful for user: admin.

46 Error Messages

46.1 On OpenERP stack trace

46.1.1 "ImportError: No module named urllib.urlopen"

Please refer to the this article, section Python 3.0 changes.

46.2 On the Swivel Log

46.2.1 AgentXML request failed, error: The agent is not authorized to access the server

User fails to authenticate with the above error message in the Swivel log.

This means that an Agent on Swivel server has not been defined for the OpenERP server.

Go to Server/Agents in the PINsafe admin console, and add a new entry, using the IP address of the OpenERP server.

Make sure the agent secret is the same as on the OpenERP configuration.

47 Oracle WebLogic

48 Overview

This document outlines the integration of Oracle WebLogic with Swivel using SAML with Swivel as an Identity Provider (IdP). It assumes that the Identity Provider and SAML Swivel Demo app are installed on the same Swivel appliance.

Swivel can provide Two Factor authentication with SMS, Token, Mobile Phone Client and strong Single Channel Authentication TURing, Pinpad or in the Taskbar.

To use the Single Channel Image such as the TURing Image, the Swivel server must be made accessible. **The client requests the images from the Swivel server, and is usually configured using a NAT (Network Address Translation)**, often with a proxy server. The Swivel virtual appliance or hardware appliance is configured with a proxy port to allow an additional layer of protection.

49 Prerequisites

Oracle WebLogic

Swivel 3.9 onwards

[Swivel AuthenticationPortal.zip](#). The file containing the IdP and login page to authenticate using Swivel.

[Swivel SAMLswivelDemo.zip](#). A simple app which sits on the service provider server to demonstrate how a user needs to be authenticated.

50 Baseline

Swivel 3.9, 3.10

Oracle WebLogic 12.1.1

51 Architecture

Swivel is configured as an Identity Provider, see the following [Oracle Documentation](#).

52 Installation

To implement the solution there are several steps:

- Setup up the Identity Provider (IdP) (Authentication Portal)
- Generate the IdP metadata (which is used to create the relationship between the IdP and Service Provider).
- Setup the service provider (the federation service and its association with the Idp)
- Create a user within PINSafe and Weblogic
- Install the demonstration application
- Test the solution

52.1 Swivel Integration Configuration

52.1.1 Configuring Swivel for Agent XML Authentication

The IdP is usually deployed on the Swivel hardware or virtual appliance, and a default localhost Agent is usually pre-configured. To make any changes to this see [Agents How to Guide](#)

52.1.2 Configuring Swivel for Single Channel Images

If Swivel Single Channel images are to be used for authentication, then the following guide can be used.

[Single Channel How To Guide](#)

52.1.3 Configuring Swivel for Dual Channel Authentication

If Swivel Dual Channel authentication methods are to be used, refer to the following guide:

[Transport Configuration](#)

52.2 Configuring the Swivel Authentication Portal

Download and extract the AuthenticationPortal.war file from the AuthenticationPortal.zip and copy this file using [WinSCP](#) to /usr/local/tomcat/webapps2 where a folder called AuthenticationPortal should appear.

Within the AuthenticationPortal folder, there will be folder called WEB-INF, with the settings.xml file (/usr/local/tomcat/webapps2/WEB-INF/settings.xml). Right click settings.xml and either Edit the file or Open in another editor such as Notepad++.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
<entry key="pinsafessl">false</entry>
<entry key="pinsafeserver">localhost</entry>
<entry key="pinsafecontext">pinsafe</entry>
<entry key="pinsafesecret">secret</entry>
<entry key="pinsafeport">8080</entry>
<entry key="imagessl">false</entry>
<entry key="imageserver">localhost</entry>
<entry key="imagecontext">pinsafe</entry>
<entry key="imageport">8080</entry>
<entry key="selfsigned">true</entry>
<entry key="serviceProviderEndpointURL">https://login.salesforce.com/?saml=02HKiPoin4nQspKPHoScmudQmsKtM.qRKnViSBCmh05IC52m5VptCNw0.p</entry>
<entry key="audience">https://saml.salesforce.com</entry>
<entry key="certificateIssuer">SAML_SP</entry>
<entry key="publicKeyFilePath">/keys/pinsafe/ssl/dsapubkey.der</entry>
<entry key="privateKeyFilePath">/keys/pinsafe/ssl/dsaprivkey.der</entry>
<entry key="certificateFilePath">/keys/pinsafe/ssl/dsacert.pem</entry>
</properties>
```

pinsafessl Communication between the IdP and Swivel. If SSL is used on the Swivel server set this to true, otherwise false. For a Swivel Hardware or Virtual appliance this should be changed to false when using port 8181 if Swivel is deployed in webapps2.

pinsafeserver Communication between the IdP and Swivel. Where the IdP is installed on the same server as Swivel this should be set to localhost.

pinsafecontext Communication between the IdP and Swivel. This is the install context and is usually pinsafe.

pinsafesecret Communication between the IdP and Swivel. By default a Swivel hardware or virtual appliance uses this value as the shared secret.

pinsafeport Communication port between the IdP and Swivel. For a Swivel Hardware or Virtual appliance this should be changed to 8181 if Swivel is deployed in webapps2 and uses a non SSL connection.

imagessl Communication between the IdP and User. If SSL is used on the Swivel server set this to true, otherwise false.

imageserver Communication between the IdP and User. If SSL is used on the Swivel server set this to true, otherwise false. By default a Swivel hardware or virtual appliance uses SSL.

imagecontext Communication between the IdP and User. This is the install context and is usually pinsafe.

imageport Communication between the IdP and User. For a Swivel Hardware or Virtual appliance this should be changed to 8443 although 443 or other port can also be used.

selfsigned Communication between the IdP and User. If SSL is used on the Swivel server with a self signed certificate then set this to true, otherwise false. By default a Swivel hardware or virtual appliance uses SSL with a self signed certificate.

serviceProviderEndpointURL the Published Site URL, defined in Setting up the Service Provider. Example:<https://192.168.10.10/saml2>

audience

certificateIssuer SAML_SP

publicKeyFilePath path to the public key usually /keys/pinsafe/ssl/dsapubkey.der

privateKeyFilePath path to the private key usually /keys/pinsafe/ssl/dsaprivkey.der

certificateFilePath path to the certificate usually /keys/pinsafe/ssl/dsacert.pem

52.3 Create private keys and certificates

Communication between Oracle and the Swivel instance is secure through the use of certificates.

52.3.1 Creating DSA Private Key

DSA key generation involves two steps, and can be done through the command line on a Swivel virtual appliance or hardware appliance:

1.

```
openssl dsaparam -out dsaparam.pem 2048
```

2.

```
openssl gendsa -out dsaprivkey.pem dsaparam.pem
```

The first step creates a DSA parameter file, dsaparam.pem, which in this case instructs OpenSSL to create a 2048-bit key in Step 2. The dsaparam.pem file is not itself a key, and can be discarded after the public and private keys are created. The second step actually creates the private key in the file dsaprivkey.pem which should be kept secret.

Export the key into a DER (binary) format. You can do so with the following steps:

1.

```
openssl dsa -in dsaprivkey.pem -outform DER -pubout -out dsapubkey.der
```

2.

```
openssl pkcs8 -topk8 -inform PEM -outform DER -in dsaprivkey.pem -out dsaprivkey.der -nocrypt
```

Step 1 extracts the public key into a DER format. Step 2 converts the private key into the pkcs8 and DER format. Once you've done this, you can use this public (dsapubkey.der) and private (dsaprivkey.der) key pair.

52.3.2 Creating a Certificate

Once you have your key pair, it's easy to create an X.509 certificate. The certificate holds the corresponding public key, along with some metadata relating to the organization that created the certificate. Follow this step to create a self-signed certificate from either an RSA or DSA private key:

```
openssl req -new -x509 -key dsaprivkey.pem -out dsacert.pem
```

After you answer a number of questions, the certificate will be created and saved as dsacert.pem.

The created keys, dsapubkey.der and dsapubkey.der need to be copied to the keys folder or wherever specified within settings.xml

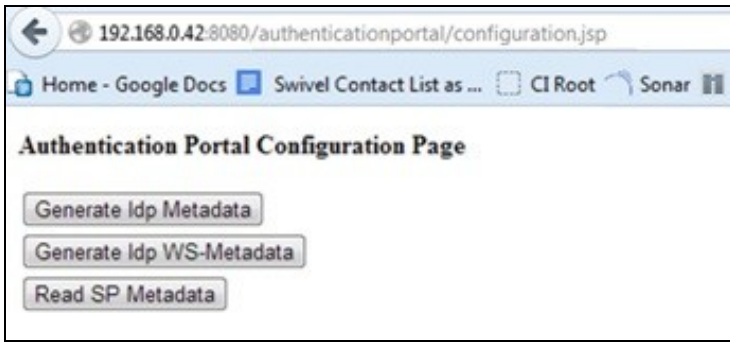
The **dsacert.pem** certificate needs to be uploaded to the GoogleApps server.

52.4 Generating IdP metadata

SAML metadata is generated by the IdP to simplify the mapping process between itself and the Service Provider.

The AuthenticationPortal folder should be located under /usr/local/tomcat/webapps2. In order to gain access to the Authentication Portal webpage, you must navigate to <https://<IPAddress>:8443/AuthenticationPortal/configuration.jsp>. (case sensitive).

This will display the configuration page as shown below. From here you should press ?Generate Idp Metadata?.



If successful, the metadata will be written to the root of the web application with the message "Metadata successfully written to" and the full path and filename displayed. Make a note of the destination which will be used later when configuring the Service Provider.

52.5 WebLogic Integration Configuration

52.5.1 Configure a WebLogic User

On the WebLogic Administration console main menu select Security Realms, select myrealm then select Users and Groups and then the Users tab to show following main screen:



Select New then input a username, this should match to the e-mail address of a Swivel user test user. Enter a dummy password as this will not be used by this integration, then press OK to save.

52.5.2 Setting up the Service Provider

On the WebLogic Administration console main menu select Environment, Servers then select AdminServer(admin). Then select Configuration, Federation Services and SAML 2.0 General to get the following screen:

Settings for AdminServer

Configuration Protocols Logging Debug Monitoring Control Deployments Services Security Notes


General Cluster Services Keystores SSL **Federation Services** Deployment Migration Tuning Overload Health Monitoring Server Sta

SAML 1.1 Source Site SAML 1.1 Destination Site **SAML 2.0 General** SAML 2.0 Identity Provider SAML 2.0 Service Provider

Save Publish Meta Data

This page configures the general SAML 2.0 per server properties

— General —

 Replicated Cache Enabled

— Site Info —

Contact Person Given Name:

Contact Person Surname:

Contact Person Type: ▼

Contact Person Company:

Contact Person Telephone Number:

Contact Person Email Address:

Organization Name:

Organization URL:

Published Site URL:

Entity ID:

Published Site URL should be your WebLogic URL + /saml2 and the Entity ID should be SAML_SP to match up other aspects of the configuration. Ensure that under the Bindings option, Recipient Check Enabled is not checked and is therefore disabled. Enter other details as appropriate then press Save.

Then, from the same screen, select SAML 2.0 Service Provider to get the following screen:

Settings for AdminServer

Configuration Protocols Logging Debug Monitoring Control Deployments Services Security Notes

General Cluster Services Keystores SSL **Federation Services** Deployment Migration Tuning Overload Health Monitoring Serv

SAML 1.1 Source Site SAML 1.1 Destination Site SAML 2.0 General SAML 2.0 Identity Provider **SAML 2.0 Service Provider**

Save

This page configures the SAML 2.0 per server service provider properties

Enabled

Always Sign Authentication Requests

Force Authentication

Passive

Only Accept Signed Assertions

Authentication Request Cache Size:

Authentication Request Cache Timeout:

POST One Use Check Enabled

POST Binding Enabled

Artifact Binding Enabled

Preferred Binding:

Default URL:

Save

Ensure the checkboxes are set as above and for the Default URL enter the path to the SAMLSwivelDemo. Press Save. Making sure that the Published Site URL is your WebLogic URL and by adding /saml2. E.g. <http://192.168.10.10/saml2> - This is your serviceProviderEndpointURL.

Going back to the section Setting up the IdP, you can go back to the settings.xml and add for example:

```
<entry key="serviceProviderEndpointURL">https://192.168.10.10/saml2</entry>
```

?

52.5.3 Specifying the IdP

On the WebLogic Administration console main menu select Security Realms, select myrealm then select Providers and Authentication to show following main screen:

Settings for myrealm

Configuration Users and Groups Roles and Policies Credential Mappings **Providers** Migration

Authentication Password Validation Authorization Adjudication Role Mapping Auditing Credential Mapping Certification Path Keystores

An Authentication provider allows WebLogic Server to establish trust by validating a user. You must have one Authentication provider in a security realm, and you can configure multiple Authentication providers in a security realm. Different types of Authentication providers are designed to access different data stores, such as LDAP servers or DBMS. You can also configure a Realm Adapter Authentication provider that allows you to work with users and groups from previous releases of WebLogic Server.

Customize this table

Authentication Providers

New Delete Reorder

Name	Description	Version
<input checked="" type="checkbox"/> DefaultAuthenticator	WebLogic Authentication Provider	1.0
<input checked="" type="checkbox"/> DefaultIdentityAsserter	WebLogic Identity Assertion provider	1.0

New Delete Reorder

Showing 1 to 2 of 2 Previous | Next

Select New to create a SAML2IdentityAsserter and name it SAML2IdentityAsserter as shown here:

Home > Providers > SAML > Summary of Security Realm > myrealm > Users and Groups > Realm Roles > Credential Mappings > Providers

Create a New Authentication Provider

OK Cancel

Create a new Authentication Provider

The following properties will be used to identify your new Authentication Provider.
* Indicates required fields

The name of the authentication provider.

* Name: SAML2IdentityAsserter

This is the type of authentication provider you wish to create.

Type: SAML2IdentityAsserter

OK Cancel

Pressing OK will take you to the following screen.

Settings for myrealm

Configuration Users and Groups Roles and Policies Credential Mappings Providers Migration

Authentication Password Validation Authorization Adjudication Role Mapping Auditing Credential Mapping Certification Path Keystones

An Authentication provider allows WebLogic Server to establish trust by validating a user. You must have one Authentication provider in a security realm, and you can configure multiple Authentication providers in a security realm. Different types of Authentication providers are designed to access different data stores, such as LDAP servers or DBMS. You can also configure a Realm Adapter Authentication provider that allows you to work with users and groups from previous releases of WebLogic Server.

Customize this table

Authentication Providers

New Delete Reorder Showing 1 to 3 of 3 Previous Next

Name	Description	Version
DefaultAuthenticator	WebLogic Authentication Provider	1.0
DefaultIdentityAsserter	WebLogic Identity Assertion provider	1.0
SAML2IdentityAsserter	SAML 2.0 Identity Assertion Provider. Supports Security Assertion Markup Language v2.0.	1.0

New Delete Reorder Showing 1 to 3 of 3 Previous Next

At this point you need to activate the changes. One way you can do this is from the main menu select Environment, select Servers then select AdminServer(admin). Then select Control. Select the checkbox next to AdminServer(admin) and Shutdown. Then restart the server and logon to the admin console.

Return to the same screen and select the SAML2IdentityAsserter.

Settings for myrealm

Configuration Users and Groups Roles and Policies Credential Mappings Providers Migration

Authentication Password Validation Authorization Adjudication Role Mapping Auditing Credential Mapping Certification Path

An Authentication provider allows WebLogic Server to establish trust by validating a user. You must have one Authentication provider in a security realm, and you can configure multiple Authentication providers in a security realm. Different types of Authentication providers are designed to access different data stores, such as LDAP servers or DBMS. You can also configure a Realm Adapter Authentication provider that allows you to work with users and groups from previous releases of WebLogic Server.

Customize this table

Authentication Providers

New Delete Reorder

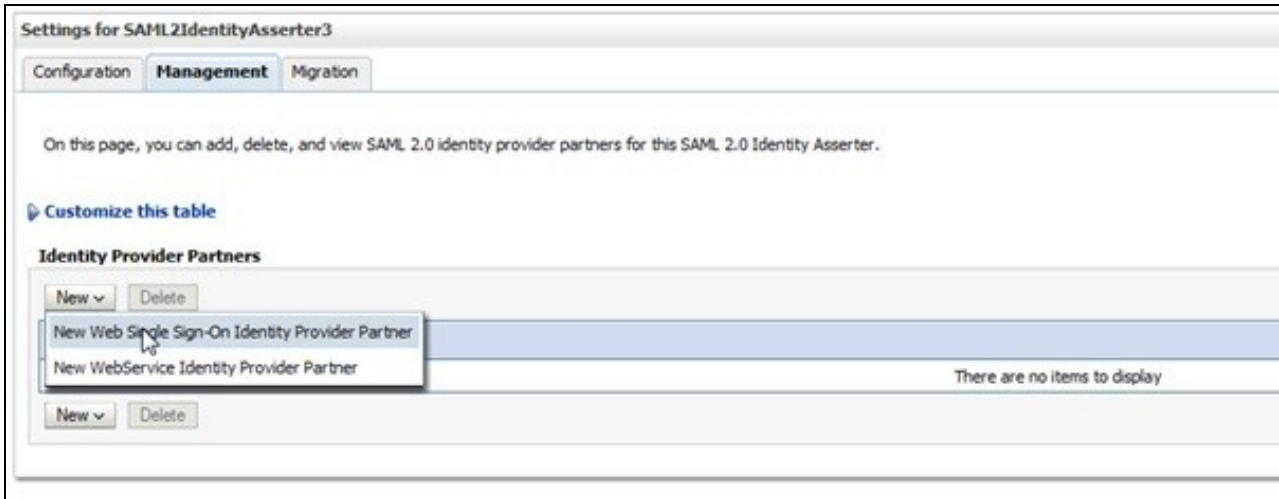
Name	Description
DefaultAuthenticator	WebLogic Authentication Provider
DefaultIdentityAsserter	WebLogic Identity Assertion provider
SAML2IdentityAsserter	SAML 2.0 Identity Assertion Provider. Supports Security Assertion Markup Language v2.0.

New Delete Reorder

Then select Management to get the screen below:



Select New and New Web Single Sign-On Identity Provider Partner as shown below:



Select New then locate and select the IdP metadata as shown below. Press OK to save

Create a SAML 2.0 Web Single Sign-on Identity Provider Partner

OK Cancel

Partner Properties

Use this page to:

- Enter the name of your new Single Sign-on Identity Provider partner
- Specify the name and location of the SAML 2.0 metadata file that you received from this new partner

* Indicates required fields

Please specify the name of the partner.

* **Name:**

Please specify the name of the file containing the partner metadata document.

Path:

Recently Used Paths:

- C:\Oracle\Middleware\user_projects\domains\base_domain
- C:\Users\adouglas\workspace3.9.2\SAMLOracle
- C:\

Current Location: 192.168.0.42 | C: | Users | adouglas | workspace3.9.2 | SAMLOracle

- .externalToolBuilders
- .settings
- .svn
- build
- src
- WebContent
- build.xml
- example.xml
- generatedIdPMetadata.xml

OK Cancel

Thus will take you to the following screen:

Settings for SAML2IdentityAsserter

Configuration **Management** Migration

On this page, you can add, delete, and view SAML 2.0 identity provider partners for this SAML 2.0 Identity Asserter.

[Customize this table](#)

Identity Provider Partners

New Delete

<input type="checkbox"/>	Name ↕
<input type="checkbox"/>	WebSSO-IdP-Partner-0

New Delete

? Select WebSSO-IdP-Partner-0 which will take you to the following screen:

Settings for SAML2IdentityAsserter

General | Site Info | Single Sign-On Signing Certificate | Transport Layer Client Certificate | Single Sign-On Service Endpoints | Artifact Resolution Service Endpoints

Save

Configures a SAML 2.0 Web Single Sign-on Identity Provider Partner's General Properties

The parameters that can be set on this Administration Console page can also be accessed programmatically via the Java interfaces that are identified in this help topic. For

— Overview —

Name: WebSSO-IdP-Partner-0

Enabled

Description:

— Authentication Requests —

Identity Provider Name Mapper Class Name:

Issuer URI: SAML_SP

Virtual User

Redirect URIs:

/SAMLswivelDemo/*

Ensure Enabled and Virtual User are checked and that Redirect URIs is set to /SAMLswivelDemo/*. Press Save to save your settings.

52.5.4 Credential Mapping Provider

On the WebLogic Administration console main menu select Security Realms, myrealm then select Providers and Authentication to show following main screen:

Settings for myrealm

Configuration | Users and Groups | Roles and Policies | Credential Mappings | **Providers** | Migration

Authentication | Password Validation | Authorization | Adjudication | Role Mapping | Auditing | **Credential Mapping** | Certification Path | Keystores

A Credential Mapping provider allows WebLogic Server to log into a remote system on behalf of a subject that has already been authenticated. You must have one Credential Mapping providers in a security realm.

Customize this table

Credential Mapping Providers

New | Delete | Reorder

Name	Description
<input type="checkbox"/> DefaultCredentialMapper	WebLogic Credential Mapping Provider

New | Delete | Reorder

Select New and then enter a name of SAML2CredentialsMapper and select type of SAML2CredentialsMapper as below (then Press OK to save):

Create a New Credential Mapping Provider

OK Cancel

Create a new Credential Mapping Provider

The following properties will be used to identify your new Credential Mapping Provider.

* Indicates required fields

The name of the Credential Mapping Provider.

* **Name:** SAML2CredentialsMap

This is the type of credential mapping provider you wish to create.

Type: SAML2CredentialMapper

OK Cancel

Select SAML2CredentialsMapper then configuration and Provider Specific. For the Issuer URI enter SAML_SP as shown below (then press Save):

Settings for SAML2CredentialsMapper

Configuration Management Migration

Common **Provider Specific**

Save

Use this page to configure provider-specific information for this SAML 2.0 Credential Mapping provider.

Issuer URI: SAML_SP

Name Qualifier:

Default Time To Live: 120

Default Time To Live Offset: -5

Web Service Assertion Signing Key Alias:

Web Service Assertion Signing Key Pass Phrase:

Please type again To confirm:

Name Mapper Class Name:

Generate Attributes

Save

52.5.5 Setting up the demo application

On the WebLogic Administration console main menu select Deployments to get the main screen looking as such:

Summary of Deployments

Control | Monitoring

This page displays a list of Java EE applications and stand-alone application modules that have been installed to this domain. Installed applications & modules are listed in the table below. To view the details of an application or module, first selecting the application name and using the controls on this page.

To install a new application or module for deployment to targets in this domain, click the Install button.

[Customize this table](#)

Deployments

Install | Update | Delete | Start v | Stop v

<input type="checkbox"/>	Name ^	State	Health	Type
There are no items to display				

Install | Update | Delete | Start v | Stop v

Select Install then locate the WAR file for the SAMLswivelDemo as such:

Install Application Assistant

Back | Next | Finish | Cancel

Locate deployment to install and prepare for deployment

Select the file path that represents the application root directory, archive file, exploded archive directory, or application module descriptor that you wish to install. The file path is displayed in the Path field.

Note: Only valid file paths are displayed below. If you cannot find your deployment files, upload your file(s) and/or confirm that your application contains the files.

Path: C:\Users\adouglas\workspace3.9.2\SAMLSwivelDemo\build\SAMLSwivelDemo.war

Recently Used Paths:

- C:\Users\adouglas\workspace3.9.2\SAMLSwivelDemo\build
- C:\Users\adouglas\workspace3.9.2\SAMLDemo\build
- C:\Users\adouglas\workspace3.9.2\SimpleDemo\build

Current Location: 192.168.0.42 \ C: \ Users \ adouglas \ workspace3.9.2 \ SAMLSwivelDemo \ build

SAMLSwivelDemo.jar

SAMLSwivelDemo.war

Back | Next | Finish | Cancel

? Click Next, Next then Finish (using all the default options) to result in the following Screen:

Messages

- ✔ All changes have been activated. No restarts are necessary.
- ✔ The deployment has been successfully installed.

Summary of Deployments

Control Monitoring


This page displays a list of Java EE applications and stand-alone application modules that have been installed to this domain, first selecting the application name and using the controls on this page.

To install a new application or module for deployment to targets in this domain, click the Install button.

Customize this table

Deployments

Install Update Delete | Start ▾ Stop ▾

<input type="checkbox"/>	Name ^
<input type="checkbox"/>	 SAML Swivel Demo

Install Update Delete | Start ▾ Stop ▾

The Demo should now be accessible.

52.6 Additional Installation Options

53 Verifying the Installation

Open a web browser and enter the URL for the root of the demo. In this case: <http://weblogicserverURL:7001/SAMLSwivelDemo>

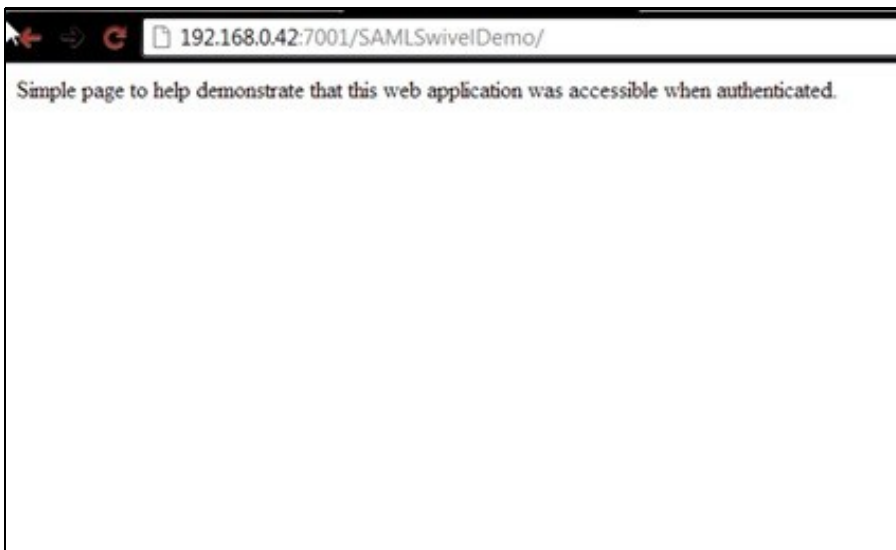
This will direct the user to the identity provider's login page as such:



A screenshot of a web browser showing the login page for SWIVEL. The browser's address bar displays the URL `192.168.0.42:8080/authenticationportal/identity_provider.jsp`. The page features the SWIVEL logo, which consists of a yellow cube icon and the text "SWIVEL the power of knowing". Below the logo, there are three input fields labeled "Username:", "Password:", and "OTC:". At the bottom of the form, there are two buttons: a blue "Login" button and a red "Refresh Image" button.

As per standard logon, enter the username and password (if required), start the session, enter the OTC and press ?Login?

If successful you will be authenticated and redirected to the SAMLDemo page as such:



54 Uninstalling the Swivel Integration

55 Troubleshooting

Check the Swivel logs

55.1 Enabling WebLogic debugging

To enable SAML logging On the WebLogic Administration console main menu select AdminServer->Configuration->Debug->Weblogic->Security->SAML2 and enable.

Now you can go to Diagnostics ->Log files ->ServerLog to view what is happening.

55.2 Error Messages

```
javax.security.auth.login.LoginException: [Security:090377]Identity Assertion Failed, weblogic.security.spi.IdentityAssertionException: [Security:090377]Identity Assertion Failed, weblogic.security.spi.IdentityAssertionException: [Security:096537]Assertion is not yet valid (NotBefore condition). at com.bea.common.security.internal.service.IdentityAssertionServiceImpl.assertIdentity(IdentityAssertionServiceImpl.java:89)
```

This has been seen where the time on the Swivel server is ahead of the WebLogics server. Ensure they both have the same time.

```
<BEA-000000> <[Security:096552]Illegal destination: https://<server_name>:<port>/saml2/sp/acs/post of assertion response.>
```

This is due to the Recipient destination value not matching the local (SP) assertion consumer URL. On the Weblogic Console => Environment => Servers => AdminServer => Configuration => Federation Services => SAM 2.0 General => disable ?Recipient Check Enabled? checkbox.

56 Known Issues and Limitations

57 Additional Information

For assistance in the Swivel installation and configuration please firstly contact your reseller and then email Swivel Secure support at support@swivelsecure.com.

58 PHP Integration

58.1 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.

58.2 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.

58.2.1 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
```

58.3 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.

58.3.1 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).

59 Swivel Combined Client

59.1 Overview

This document describes a combined authentication and administration client for Swivel Server APIs which can be used in web servers using ASP.Net. It is a convenient wrapper around the XML-based APIs described [here](#).

The client can also be used in non-web applications using the .Net Framework. See configuration below.

59.2 Prerequisites

The client DLL is available from [here](#). This is version 1.3 of the client, which includes support for user attributes and for TLS protocol versions 1.1 and 1.2. This requires version 3.9.7 or later of the Swivel server. For TLS 1.1/1.2 support, version 3 of the Swivel appliance is required.

An example website demonstrating some of the features of the client is available from [here](#). This download includes the DLL above, so you don't need to download both. Again, this test server uses the new client.

This version of the client requires Microsoft.Net Framework version 4.5 or later.

59.3 Using the Client

59.3.1 Configuring The Swivel Server

In order to use the client with a particular Swivel Server, you must define the computer running your web application as an Agent on that server. To do this:

- Log into the Swivel Server Administration Console
- Go to Server -> Agents
- In version 3.8 or later, expand the entry at the bottom labelled "New Entry". In earlier versions, this will already be visible.
- Enter a name for the new Agent. Note that any users created using the API will be added to a repository with this name, so the name should not be the same as an existing repository if you intend to manage users through this Agent.
- Enter the IP address of the web server under Host/IP.
- If you will be managing users through this Agent, set "Can act as Repository" to Yes.
- The remaining entries can be left as default for now. Click Apply.

Depending on how you intend to use the client, you may like to create a new Repository Group for users created by the Agent, or you may prefer to use an existing Group. You can create a new Group under Repository -> Groups. All you need to add is a name in the blank entry at the bottom. There is no need to add definitions for other repositories, or to specify rights for this Group, as you will be specifying user rights explicitly when the user is created. You may also like to go back to the Agent definition and set the Group for this Agent to the Group you have just created. If you do this, only users created by the Agent will be able to authenticate through the Agent.

If you have created a new Group, you should also set up Transports for this Group. You should at least set up an Alert Transport, and if you are using dual channel authentication, a Strings Transport as well. Note that if you want to use an existing Transport, you will need to copy the class name to a new Transport and give it a new name. Select the Group you have just created as the Strings or Alert Group as appropriate.

59.3.2 Configuring Your ASP.Net Application

In order to use the client, you need to add certain entries to the web.config file in your ASP.Net application. The following is an example:

```
<appSettings>
  <add key="PINsafeServer" value="myserver"/>
  <add key="PINsafePort" value="8080"/>
  <add key="PINsafeContext" value="sentry"/>
  <add key="PINsafeSecret" value="secret"/>
  <add key="PINsafeSecure" value="True"/>
  <add key="PINsafeAcceptSelfSigned" value="False"/>
  <add key="AllowNonPINsafeUsers" value="False"/>
  <add key="IgnoreDomainPrefix" value="True"/>
  <add key="IgnoreDomainSuffix" value="False"/>
  <add key="PINsafeAgentVersion" value="3.97" />
  <add key="PINsafeTlsProtocols" value="Tls12" />
</appSettings>
```

The comments at the start and end are not necessary - they are for clarity - but there is a method in the client to remove these settings which requires these comments. The <appSettings> element should be contained within the main <configuration> element, and may already exist, depending on your application.

If you are using the client in an executable (i.e. non-web) application, you can use exactly the same settings, but these must be in a file named *Application.exe.config*, where *Application* is the name of the executable application. This must be in the same directory as the program executable.

The meanings of the key names are shown below:

- **PINsafeServer** - The host name or IP address of the Swivel server
- **PINsafePort** - The port used by the Swivel server, normally 8080
- **PINsafeContext** - The context (application URL) of the Swivel server, normally "pinsafe"
- **PINsafeSecret** - The shared secret as configured in the Agent entry previously
- **PINsafeSecure** - "True" if HTTPS is to be used for communication with, "False" otherwise
- **PINsafeAcceptSelfSigned** - "True" if SSL certificate errors should be ignored
- **AllowNonPINsafeUsers** - "True" if users unknown to PINsafe should be authenticated automatically, "False" if they should be rejected
- **IgnoreDomainPrefix** - "True" if the domain prefix in usernames such as domain\user should be stripped off before checking with Swivel server
- **IgnoreDomainSuffix** - "True" if the domain suffix in usernames such as user@domain should be stripped off before checking with Swivel server
- **PINsafeAgentVersion** - Sets the *Version* attribute sent with the API request. The default is "3.4". To support user attributes, this should be set to "3.97".

- **PINsafeTlsProtocols** - Specifies which TLS protocols are supported. You can specify Ssl3, Tls1, Tls11 or Tls12. Separate supported protocols with commas. The default is "Tls1,Tls11,Tls12".

NOTE: on an appliance, you should not use the proxy application (and port 8443), as functionality in the proxy is deliberately limited. You must use the pinsafe application to get the full functionality out of the client.

If it is not practical to provide the settings in a configuration file, you can create an instance of the SwivelSettings class, and initialise it as follows:

```
SwivelSettings swivelSettings = SwivelSettings.EmptySettings;
swivelSettings.Server = "myserver";
swivelSettings.Port = 8080;
swivelSettings.Context = "sentry";
swivelSettings.Ssl = true;
swivelSettings.Secret = "secret";
swivelSettings.AcceptSelfSigned = false;
swivelSettings.TlsProtocols = SecurityProtocolType.Tls12;
```

You can then use this instance when creating a request - see the class documentation for details.

59.3.3 Implementing the Swivel Client in ASP.Net Applications

In order to use the client DLL, you need to copy it to the Bin folder of your ASP.Net application.

Technical documentation for the client is in progress. Please see the example application.

The namespace for all classes in the client is **swivelsecure.client**.

Documentation of the authentication class can be found [here](#)

The user management classes fall into two categories: [administrator](#) and [helpdesk](#). The administrator methods are more extensive, allowing you to create, update and delete users. However, they are limited to users belonging to a single repository, named after the Agent corresponding to the computer the application is running on. The helpdesk methods are more limited, typically read-only methods, but with some update functionality. However, these methods can apply to users in other repositories. See the two documents for details.

60 Website Integration

61 Website Authentication

There are a number of ways of authenticating custom-built websites and web based applications, depending on the nature of the application and how much development you can do or are prepared to do.

- A XML-based API for authentication is provided, see [Agent-XML](#).
- For ASP.Net Swivel has a DLL wrapper around this API. See, [Swivel_Combined_Client](#).
- For IIS, see: [Microsoft_IIS_version_7_Integration](#).
- For IIS forms based authentication see: [Microsoft_IIS_version_7_ASP.NET_Forms_Integration](#).
- For a PHP wrappers for the API: see [PHP_Integration](#).
- For a Java wrappers for the API: see [Java_Client](#).