

Sitecore CMS 7.0 or later

Sitecore Security Hardening Guide

Recommendations for making Sitecore more secure

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2014 Sitecore. All rights reserved.

Table of Contents

Chapter 1 Introduction 3
Chapter 2 Security Settings 4
2.1 General Security Information 5
2.2 Limiting Access to .XML, .XSLT, and .MRT Files 6
2.3 Protecting Folders in the IIS 7
2.3.1 Limiting Anonymous Access to Folders 7
2.4 The Structure of the Website Folder 9
2.5 Turn off Auto Complete of Username in the Login Page 10
2.6 Making the Sitecore Login Page Available to SSL Requests Only 11
2.7 Controlling File Upload 12
2.7.1 Deny Execute Permissions on the Upload Folder 12
Denying Execute Permission in IIS 12
2.7.2 Disabling the Upload Watcher 13
2.7.3 The Upload Filter Tool 14
Installing the Upload Filter Tool 14
Configuring the Upload Filter Tool 14
2.8 Security and Client RSS Feeds 16
2.8.1 Disabling Client RSS Feeds 16
2.9 Removing Headers from Responses 17
2.9.1 Removing the X-Aspnet-Version HTTP Header 17
2.9.2 Removing the X-Powered-By HTTP Header 17
2.9.3 Removing the X-AspNetMvc-Version HTTP Header 17
2.10 Recommended Reading 18
2.10.1 Other Resources 18

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2014 Sitecore. All rights reserved.

Chapter 1

Introduction

The Security Hardening Guide is designed to help you make your Sitecore installation as secure as possible.

Sitecore is of course subjected to rigorous testing before each release and any bugs or security threats that may exist are fixed and removed as soon as they are discovered. We also release updates whenever necessary.

However, the way you implement your Sitecore installation has a significant effect on the security of your website.

This document contains details of our best practices and recommendations for ensuring that your Sitecore installation is as secure as possible.

Sitecore is not responsible for the security of any other software products that you use with your website. We strongly recommend that you install *every* available service pack and update for *all* of the software products that you use.

It is important to remember that secure software is a goal that we are constantly trying to achieve but may never reach.

Security is risk management; it is about understanding the risks and concrete threats to your environment and mitigating against them. You must analyze the threats and risks that your installation faces and then do your utmost to secure your installation against these threats.

This document does not describe the Sitecore Security system. For more information about the Sitecore security system, see the *Security Administrators Cookbook*.

This Security Hardening Guide contains the following chapters:

- Introduction
- Security Settings

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2014 Sitecore. All rights reserved.

Chapter 2

Security Settings

This chapter describes the settings that you should apply in your Sitecore installation.

This chapter contains the following sections:

- General Security Information
- Limiting Access to .XML, .XSLT, and .MRT Files
- Protecting Folders in the IIS
- The Structure of the Website Folder
- Turn off Auto Complete of Username in the Login Page
- Making the Sitecore Login Page Available to SSL Requests Only
- Controlling File Upload
- Security and Client RSS Feeds
- Removing Headers from Responses
- Recommended Reading

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2014 Sitecore. All rights reserved.

2.1 General Security Information

Although Sitecore can run on several different operating systems, we recommend that you use the newest operating systems with the most up-to-date security features. Use the Windows update / Automatic update service to keep all your client computers and servers up-to-date with the most recent security updates and service packs.

You should also create a disaster recovery plan to ensure the rapid resumption of services should a disaster occur. The recovery program should include:

- A plan for acquiring new or temporary equipment.
- A plan for restoring backups.
- Testing the recovery plan.

When you use the installation program to install Sitecore, all of the appropriate security settings are set. However, if you install Sitecore from a `.zip` file or if you install a website on a server without running the `setup.exe`, there are a number of settings that you will have to set manually. These settings are described in detail in the *Sitecore CMS 7.0 Installation Guide* in sections 4.2 to 4.3.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2014 Sitecore. All rights reserved.

2.2 Limiting Access to .XML, .XSLT, and .MRT Files

To improve the security of your Sitecore installation, you must edit the `web.config` file. This file is stored in the `\WebSite` folder of you installation, for example at `C:\Inetpub\wwwroot\YourWebsite\WebSite`

To limit access to `.XML`, `.XSLT`, and `.MRT` files:

1. Open the `web.config` file.
2. Add the following lines to the `<system.webServer><handlers>` section:

```
<system.webServer>
  <handlers>

    <!-- Add managed handler for IIS Classic Mode in order to prevent access to files
         Notice: Must correspond to the handlers defined in <httpHandlers> section -->
    <add path="*.xml" name="xml Handler (classic)" verb="*" modules="IsapiModule"
scriptProcessor="%windir%\Microsoft.NET\Framework\ v4.0.30319\aspnet isapi.dll"
resourceType="Unspecified" preCondition="classicMode,runtimeVersionv4.0" />
    <add path="*.xslt" name="xslt Handler (classic)" verb="*" modules="IsapiModule"
scriptProcessor="%windir%\Microsoft.NET\Framework\ v4.0.30319\aspnet_isapi.dll"
resourceType="Unspecified" preCondition="classicMode,runtimeVersionv4.0" />
    <add path="*.config.xml" name="config.xml handler (classic)" verb="*" modules="IsapiModule"
scriptProcessor="%windir%\Microsoft.NET\Framework\ v4.0.30319\aspnet isapi.dll"
resourceType="Unspecified" preCondition="classicMode,runtimeVersionv4.0" />
    <add path="*.mrt" name="mrt handler (classic)" verb="*" modules="IsapiModule"
scriptProcessor="%windir%\Microsoft.NET\Framework\ v4.0.30319\aspnet_isapi.dll"
resourceType="Unspecified" preCondition="classicMode,runtimeVersionv4.0" />

    <!-- Prevent files from being served in IIS Integrated Mode -->
    <add path="*.xml" verb="*" type="System.Web.HttpForbiddenHandler" name="xml (integrated)"
preCondition="integratedMode"/>
    <add path="*.xslt" verb="*" type="System.Web.HttpForbiddenHandler" name="xslt (integrated)"
preCondition="integratedMode"/>
    <add path="*.config.xml" verb="*" type="System.Web.HttpForbiddenHandler" name="config.xml
(integrated)" preCondition="integratedMode"/>
    <add path="*.mrt" verb="*" type="System.Web.HttpForbiddenHandler" name="mrt (integrated)"
preCondition="integratedMode"/>
```

3. Add the following lines to the `<system.web><httpHandlers>` section:

```
<system.web>
  <httpHandlers>

    <!-- Prevent files from being served in IIS Classic Mode -->
    <add path="*.xml" verb="*" type="System.Web.HttpForbiddenHandler" validate="true" />
    <add path="*.xslt" verb="*" type="System.Web.HttpForbiddenHandler" validate="true" />
    <add path="*.config.xml" verb="*" type="System.Web.HttpForbiddenHandler" validate="true" />
    <add path="*.mrt" verb="*" type="System.Web.HttpForbiddenHandler" validate="true" />
```

Sitecore under Windows x64

If Sitecore is running under Windows x64, you must set the scriptProcessor attribute to the Framework64 folder: `scriptProcessor="%windir%\Microsoft.NET\Framework64\....."` .

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2014 Sitecore. All rights reserved.

2.3 Protecting Folders in the IIS

You can improve security by preventing anonymous users from accessing certain key folders.

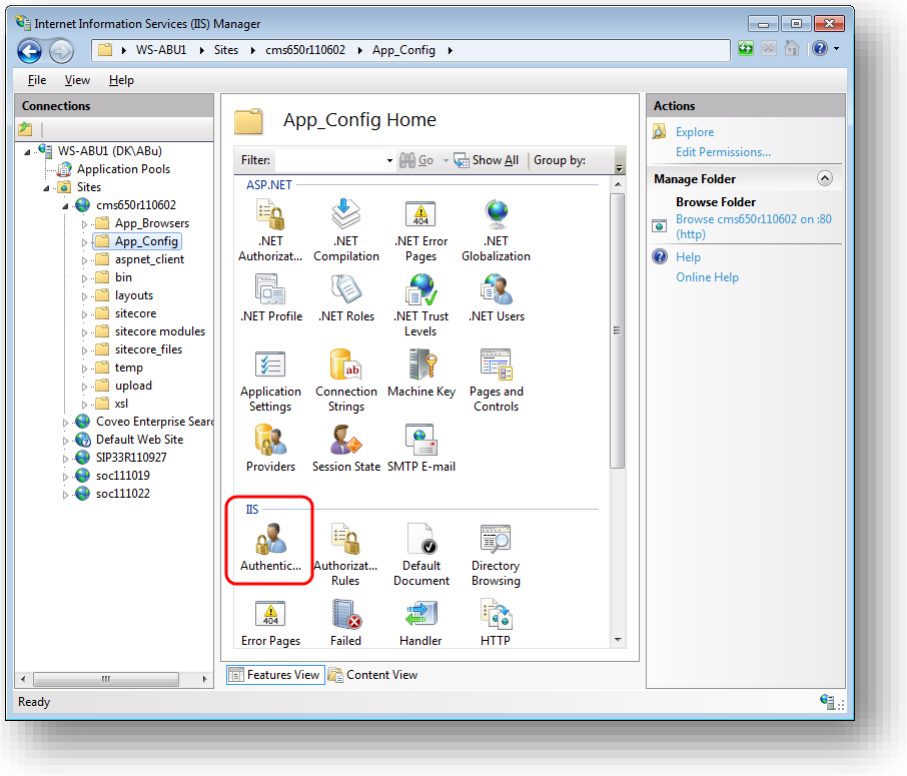
You should prevent anonymous users from accessing the following folders:

- */App_Config*
- */sitecore/admin*
- */sitecore/debug*
- */sitecore/shell/WebService*

2.3.1 Limiting Anonymous Access to Folders

To limit anonymous access to the /`App_Config` folder:

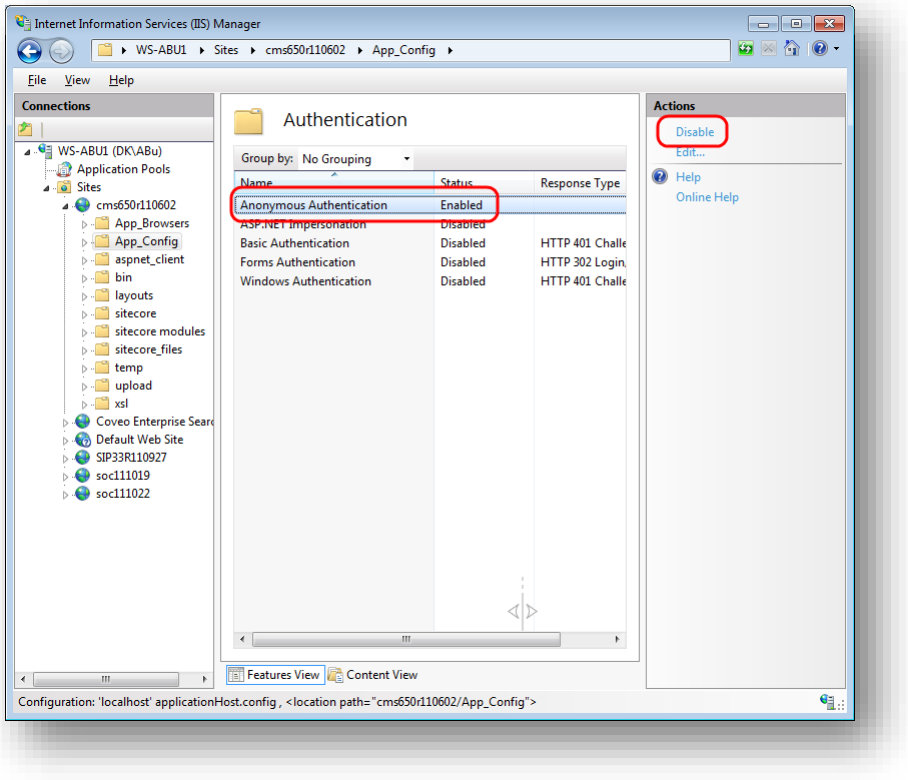
1. Open the IIS.
1. Navigate to the `Web Sites\Default Web Site\App_Config` folder.



2. In **Features View**, double-click **Authentication**.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2014 Sitecore. All rights reserved.

3. In the **Authentication** window, select **Anonymous Authentication** and in the **Actions** panel, click **Disable**.



4. Restart IIS.

Repeat this process for the `/sitecore/admin`, `/sitecore/debug` and `/sitecore/shell/WebService` folders.

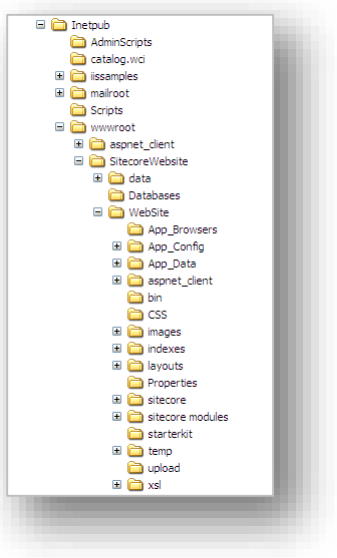
Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2014 Sitecore. All rights reserved.

2.4 The Structure of the Website Folder

You can improve security by placing the following folders outside the website root folder:

- */data*
- */indexes*

After moving the `/data` folder you must edit the `web.config` file to point to the new location. You must also configure permissions for ASP.NET requests. For more information about that, see section *4.2.2 File System Permissions for ASP.NET Requests* in the *CMS 7.0 Installation Guide*.



You can install Sitecore using:

- The installation program.
- A `.zip` file.

Using the Installation Program

If you use the installation program to install Sitecore, the `/data` folder is created outside the website root folder and the `web.config` file is edited to point to this location. The `/indexes` folder is placed in the `/data` folder.

This is the recommended configuration and you don't need to make any changes.

Using a .zip File

If you use a `.zip` file to install Sitecore, the data folder is created outside the website root folder, but the `web.config` file is not edited to point to this location. The `/indexes` folder is placed in the `/data` folder. When you run Sitecore for the first time, it creates another data folder in the `/WebSite` folder.

We therefore recommend that you edit the `web.config` file to point to the correct location.

The `web.config` file should look like this:

```
        <sitecore database="SqlServer">
          <sc.variable name="dataFolder" value="C:\Inetpub\wwwroot\SitecoreWebsite\data\" />
          <sc.variable name="mediaFolder" value="/upload" />
          <sc.variable name="tempFolder" value="/temp" />
```

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2014 Sitecore. All rights reserved.

2.5 Turn off Auto Complete of Username in the Login Page

You can also improve the security of your Sitecore installation by specifying that Sitecore should not automatically complete the username of users when they log in.

To turn off the auto-completion of user names:

1. Navigate to the `C:\Inetpub\wwwroot\YourWebsite\WebSite\sitecore\login` folder.
2. Open the `default.aspx` file.
3. Locate the `form id="LoginForm"` section
4. Edit this section as follows:

```
<form id="LoginForm" runat="server" autocomplete="off">
```

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2014 Sitecore. All rights reserved.

2.6 Making the Sitecore Login Page Available to SSL Requests Only

You can configure the Sitecore CMS to use only SSL requests for the Sitecore login page.

Create a custom redirect processor that will redirect from *http://hostname/sitecore/login* to *https://hostname/sitecore/login*, and redirect all other pages from *https* to *http*.

Use the following code as an example:

```
        public class SslLogin
            {
                public void Process(PipelineArgs args)
                {
                    string absUrl = HttpContext.Current.Request.Url.AbsoluteUri;
                    string localUrl = HttpContext.Current.Request.Url.LocalPath;

                    if (localUrl.StartsWith("/sitecore/login") && absUrl.StartsWith("http://")
&& !Context.IsLoggedIn)
                    {
                        HttpContext.Current.Response.Redirect(absUrl.Replace("http://",
"https://"));
                        return;
                    }
                    if (!localUrl.StartsWith("/sitecore/login") && absUrl.StartsWith("https://")
&& Context.IsLoggedIn)
                    {
                        HttpContext.Current.Response.Redirect(absUrl.Replace("https://",
"http://"));
                    }
                }
            }
```

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2014 Sitecore. All rights reserved.

2.7 Controlling File Upload

You can strengthen security of your Sitecore installation by controlling access to files that are uploaded by the users.

2.7.1 Deny Execute Permissions on the Upload Folder

If you allow users to modify the content of the /`upload` folder, you also give them the permission to place scripts and executable programs in the folder. Executing these scripts and programs can cause an unexpected behavior on the server. You must therefore prevent an uploaded file from being executed on the server side when a user attempts to download it.

We recommend that you deny permissions to run scripts and executable files in the `/upload` folder.

Note
You only need to perform this step if your configuration allows content authors to place files directly to the `/upload` folder. For example, if you use a shared directory or FTP server, content authors can quickly place a lot of media in the media library.

For more information about *Execute* permission in IIS, see http://support.microsoft.com/kb/313075.

Deny uploading files to the Temp folder

Deny users to upload files to the `/temp` folder as well.

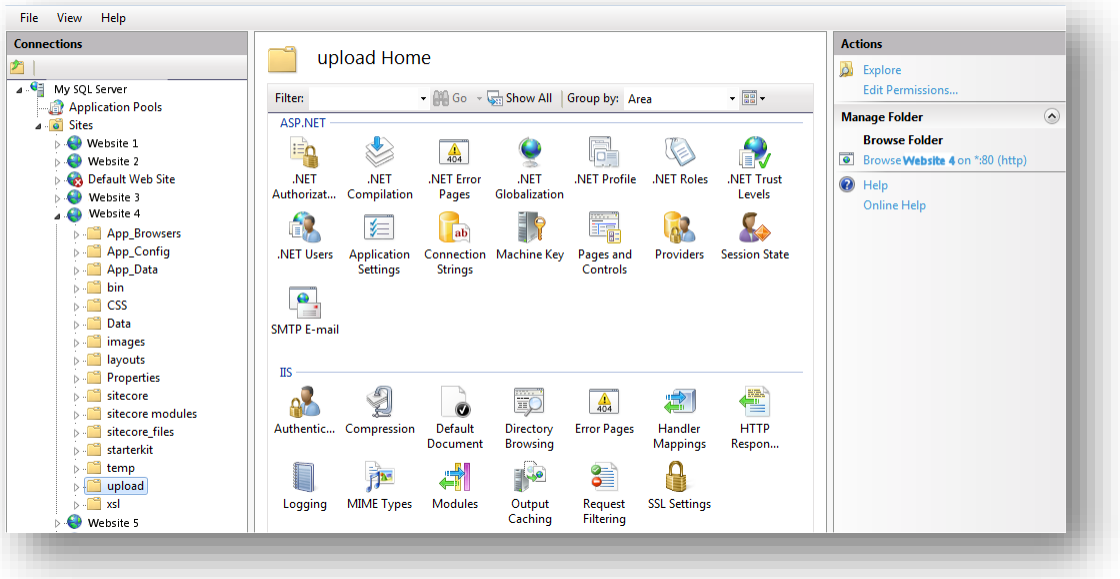
Note
This step is primarily needed if your configuration allows content authors to place files directly to the `/temp` folder using a shared directory or a FTP server. But we recommend that you perform this step in any case to avoid potential security problems if `.aspx` files for some reason end up being saved in the `/temp` folder (for example from custom code).

Denying Execute Permission in IIS

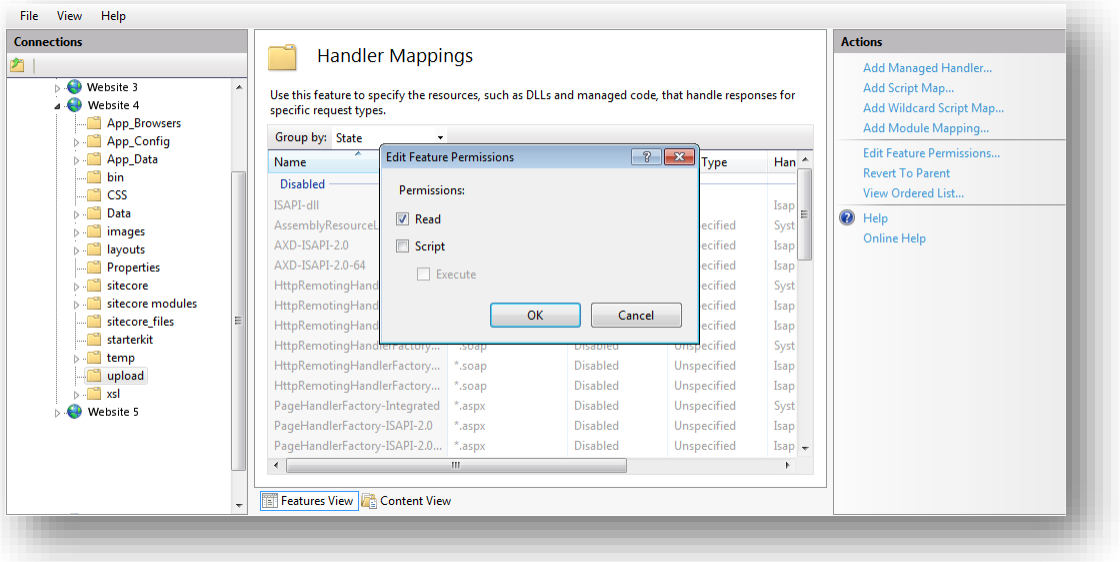
You must deny both *Script* and *Execute* permission to the upload folder.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2014 Sitecore. All rights reserved.

1. Navigate to the `/upload` folder for the database that you are interested in.



2. Select the `/upload` folder and click **Handler Mappings** and then in the **Actions** pane, click **Edit Feature Permissions**.



3. In the **Edit Feature Permissions** dialog box, clear the **Script** and **Execute** check boxes.

2.7.2 Disabling the Upload Watcher

We recommend that you disable **Upload Watcher** and thereby ensure that the only way to upload files is from the **Media Library**. This ensures that you can only upload files from within the Sitecore client and have control over the files that are uploaded.

When **Upload Watcher** is disabled, files that are placed in the `/upload` folder are not automatically uploaded to the **Media Library**.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2014 Sitecore. All rights reserved.

To disable the **Upload Watcher**, remove the following line from the `<modules>` section of the `Web.config` file.

```
  <system.webServer>
    <modules>
      <remove name="ScriptModule"/>
      <add type="Sitecore.Nexus.Web.HttpModule,Sitecore.Nexus" name="SitecoreHttpModule"/>
      <add type="Sitecore.Resources.Media.UploadWatcher, Sitecore.Kernel"
name="SitecoreUploadWatcher"/>
```

2.7.3 The Upload Filter Tool

However, if you want to have complete control and prevent users from uploading certain file types, you should use the **Upload Filter** tool.

The **Upload Filter** tool lets you prevent certain file types from being uploaded, for example `.exe`, `.dll`, and so on.

You can download the Upload Filter tool — `Upload Filter-1.0.0.2.zip` — from the Sitecore Developer Network where it is available as a Sitecore package file along with this Security Hardening Guide.

The Sitecore package contains the following files:

File Name	Destination Folder
`UploadFilter.config`	`Website\App_Config\Include\`
`UploadFilter.dll`	`WebSite\bin\`

Installing the Upload Filter Tool

You must install the Sitecore package file before you can use the **Upload Filter** tool.

To install Upload Filter tool:

1. In the **Sitecore Desktop**, click **Sitecore**, **Control Panel**.
2. In the **Sitecore Control Panel**, click **Administration**, **Install a Package**.
3. The wizard will guide you through the installation process.

Configuring the Upload Filter Tool

After you install the package, you must configure the tool.

To configure the **Upload Filter** tool:

4. Open the `UploadFilter.config` file.

```
<processors>
            <uiUpload>
                <processor mode="on" type="Sitecore.Pipelines.Upload.CheckExtension,
Sitecore.UploadFilter" patch:before="*[1]">
                    <param desc="Allowed extensions (comma separated)"></param>
                    <param desc="Blocked extensions (comma separated)">exe,dll</param>
                </processor>
            </uiUpload>
        </processors>
```

5. In the `Allowed extensions` parameter, enter a comma-separated list of the file extension types that can be uploaded.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2014 Sitecore. All rights reserved.

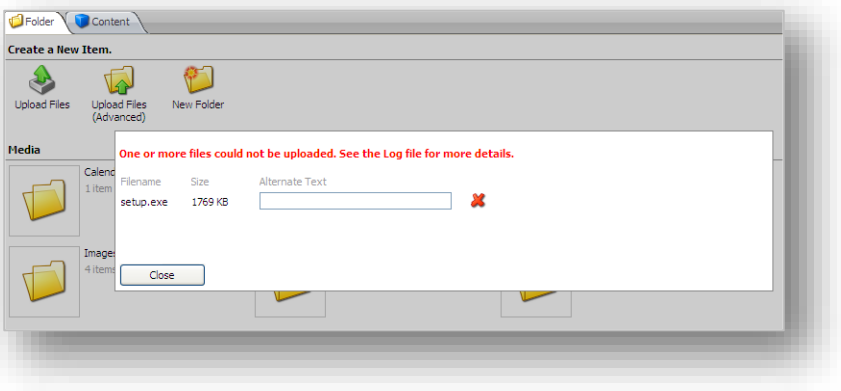
Or

In the `Blocked extensions` parameter, enter a comma-separated list of the file extension types that cannot be uploaded.

You must enter the file extension without the dot.

Important
If you set the `Allowed extensions` parameter, the `Blocked extensions` parameter is ignored.

6. If you try to upload a file type that is on the blocked list, you see the following message:



Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2014 Sitecore. All rights reserved.

2.8 Security and Client RSS Feeds

RSS technology is designed so that users who follow an RSS link can come directly to the item specified in the URL of the RSS feed. Most RSS readers do not support authentication. This means that users who subscribe to Sitecore client RSS feeds have direct access to the item specified in the URL of the RSS feed and do not have to identify themselves to the Sitecore security system when they view the RSS feed. However, the Sitecore security system verifies that they are authorized users when they try to perform any actions associated with the client feed.

If someone else gains access to the URL of the RSS feed:

- They *can* follow the link and view all the content contained in the RSS feed even though their own security permissions do not give them access to this item.
- They *cannot* perform any actions on the content.
- They *cannot* view any other content.
- They *cannot* gain access to the username or password of the original owner of the RSS feed.
- They *cannot* modify the link to gain access to any other content.

Important
Sitecore users should not share RSS feeds.

2.8.1 Disabling Client RSS Feeds

If your Sitecore installation contains sensitive information that you want to protect, you can disable Sitecore client RSS feeds.

To disable Sitecore client feeds:

1. Open the `web.config` file.
2. Locate the `<httpHandlers>` section. Depending on your IIS pool, this section may be called Handlers.
3. Remove the following handler:

```
<add verb="*" path="sitecore_feed.ashx"
type="Sitecore.Shell.Feeds.FeedRequestHandler, Sitecore.Kernel"/>
```

Removing this handler disables all the client feeds that are available inside Sitecore. However, any public RSS feeds that you have created are still available to Website visitors.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2014 Sitecore. All rights reserved.

2.9 Removing Headers from Responses

You can improve security and save a small amount of bandwidth by removing header information from each response sent by your web site.

These headers contain a number of infrastructure details about the framework used on your website that you do not need to publicize.

You can easily remove:

- The X-Aspnet-Version HTTP header.
- The X-Powered-By HTTP header.
- The X-AspNetMvc-Version HTTP header.

2.9.1 Removing the X-Aspnet-Version HTTP Header

By removing the X-Aspnet-Version HTTP header information from each web page, you save a little bandwidth and ensure that you are not publicizing which version of ASP.NET you are using.

To remove the X-Aspnet-Version HTTP header from each response from ASP.NET, add the following to the `web.config` file.

```
<system.web>
  <httpRuntime enableVersionHeader="false" />
</system.web>
```

For more information about removing the X-Aspnet-Version HTTP header, see http://www.dotnetperls.com/x-aspnet-version.

2.9.2 Removing the X-Powered-By HTTP Header

By removing the X-Powered-By HTTP header, you are not publicizing which version of ASP.NET you are using.

To remove the X-Powered-By HTTP header from each response from ASP.NET, add the following to the `web.config` file.

```
<system.webServer>
  <httpProtocol>
    <customHeaders>
      <remove name="X-Powered-By" />
    </customHeaders>
  </httpProtocol>
</system.webServer>
```

2.9.3 Removing the X-AspNetMvc-Version HTTP Header

By removing the X-AspNetMvc-Version HTTP header, you are not publicizing which version of ASP.NET MVC you are using.

To remove the X-AspNetMvc-Version HTTP header, add the following to the `Application_Start` method in the `Global.asax.cs` file:

```
protected void Application_Start(object sender, EventArgs e)
{
  MvcHandler.DisableMvcResponseHeader = true;
  // RegisterRoutes etc... and other stuff
}
```

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2014 Sitecore. All rights reserved.

2.10 Recommended Reading

For information about how to make Sitecore more secure, we recommend that you read the following documents:

- Sitecore Installation Guide.

  http://sdn.sitecore.net/Products/Sitecore%20V5/Sitecore%20CMS%207/Installation.aspx
- The upgrade instructions for the Sitecore version you are upgrading from.
- The `web.config` changes documentation for the Sitecore version you are upgrading to.

2.10.1 Other Resources

For more information about SQL Server security, visit:

http://technet.microsoft.com/en-us/library/bb545450.aspx

http://www.microsoft.com/en-us/sqlserver/solutions-technologies/mission-critical-operations/security-and-compliance.aspx

For more information about security in general, visit the Microsoft Security TechCenter:

http://technet.microsoft.com/en-us/security/default.aspx

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2014 Sitecore. All rights reserved.