Sitecore CMS 7.0 以降

Sitecore セキュリティ強化ガイド

Sitecore インストール環境のセキュリティを向上させるための推奨事項

目次

Chapter 1

イントロダクション

セキュリティ強化ガイドは、Sitecore インストール環境を最大限安全にすることを意図したものです。

Sitecore は、各リリース前に既に厳しい検査を受けており、存在する可能性のあるバグまたはセキュリティ上の脅威は発見次第修正および削除しています。また、必要に応じて更新プログラムも随時リリースしています。

そこで、Web サイトのセキュリティに大きな影響を及ぼすのは、Sitecore インストールの実行方法です。

この文書では、Sitecore インストール環境を最大限安全にするためのベスト プラクティスおよび推奨事項について、詳細に説明しています。

Sitecore では、お客様の Web サイトで使用された他社のソフトウェア製品のセキュリティについては責任を負いません。すべての入手可能なサービス パックをインストールし、使用するソフトウェア製品をすべてアップデートすることを強くお勧めします。

ソフトウェアを安全な状態に保つには、絶え間ない努力が必要であり、完全に安全ということはほぼあり得ないということをご理解ください。

セキュリティはリスク管理です。ご使用の環境に対するリスクや現実の脅威を理解し、それらを軽減することです。インストールにおいて直面する脅威やリスクを分析し、これらの脅威に対してインストール環境の安全を確保するよう最善を尽くす必要があります。

この文書は、Sitecore セキュリティ システムについては説明しません。Sitecore セキュリティ システムの詳細については、『セキュリティ管理者クックブック』を参照してください。

このセキュリティ強化ガイドには次の章があります。

- イントロダクション
- セキュリティ設定

Chapter 2

セキュリティ設定

この章では、Sitecore インストール環境に適用が必要ないくつかの設定について説明します。

この章には次のセクションがあります。

- 一般セキュリティ情報
- .XML、.XSLT および .MRT ファイルへのアクセス制限
- IIS でのフォルダーの保護
- Web サイト フォルダーの構造
- ログイン ページでのユーザー名のオート コンプリートの無効化
- ファイルのアップロードの制御
- セキュリティおよびクライアント RSS フィード
- 応答からのヘッダーの削除
- 参考文献

2.1 一般セキュリティ情報

Sitecore は複数の異なるオペレーティング システム上で実行可能ですが、オペレーティング システムを最新のセキュリティ機能を備えた状態で使用することをお勧めします。Windows Update/自動更新のサービスを使用して、クライアント コンピューターおよびサーバーすべてを最新のセキュリティ アップデートおよびサービス パックで最新の状態にするようにしてください。

また、災害復旧計画を作成して、災害発生時にはサービスを早急に再開できるようにしておく必要があります。復旧計画には次を含める必要があります。

- 新しい機器または一時的に使用する機器を調達する計画
- バックアップを復元する計画
- 復旧計画のテスト

インストール プログラムを使用して Sitecore をインストールする場合、適切なセキュリティ設定もすべて行われます。ただし、.zip ファイルから Sitecore をインストールする場合、または setup.exe を実行せずにサーバー上に Web サイトをインストールする場合、いくつかの設定を手動で行う必要があります。これらの設定については、『Sitecore CMS 7.0 Installation Guide』のセクション 4.2 から 4.3 に詳細に説明されています。

2.2 .XML、.XSLT および .MRT ファイルへのアクセス制限

Sitecore インストール環境のセキュリティを向上させるには、web.config ファイルを編集する必要があります。このファイルは、インストール環境の \WebSite フォルダーに保存されています。たとえば、

C:\Inetpub\wwwroot\YourWebsite\WebSite です。

.XML、.XSLT および .MRT ファイルへのアクセスを制限する方法:

1. web.config ファイルを開きます。
2. 次の行を <system.webServer><handlers> セクションに追加します。

```
<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. 次の行を <system.web><httpHandlers> セクションに追加します。

```
<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" />
```

Windows x64 で動作する Sitecore

Sitecore が Windows x64 で動作する場合は、scriptProcessor 属性を Framework64 に設定する必要があります。フォルダー: scriptProcessor="%windir%\Microsoft.NET\Framework64\....." .

2.3 IIS でのフォルダーの保護

匿名ユーザーが特定の主要フォルダーにアクセスするのを阻止して、セキュリティを向上させることができます。

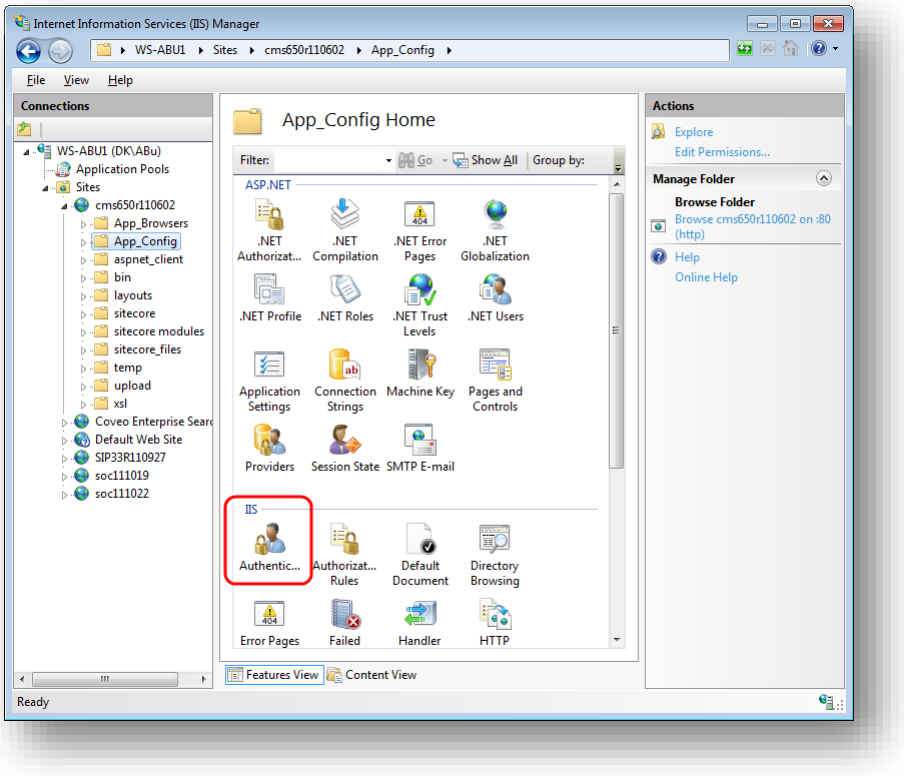
匿名ユーザーが次のフォルダーにアクセスするのを阻止する必要があります。

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

2.3.1 フォルダーへの匿名アクセスの制限

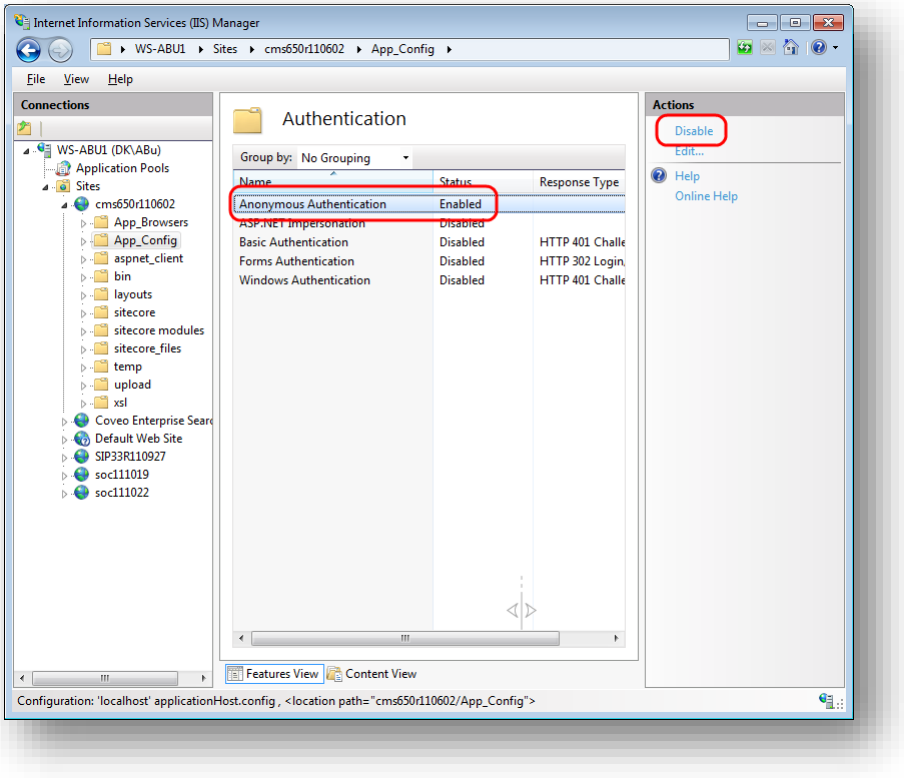
`/App_Config` フォルダーへの匿名アクセスを制限する方法

1. IIS を開きます。
2. `Web Sites\Default Web Site\App_Config` フォルダーに移動します。



3. [**機能ビュー**] で、[**認証**] をダブルクリックします。

4. [**インターネット インフォメーション サービス**] ウィンドウで `App_Config` フォルダーを右クリックして [プロパティ] をクリックします。



5. IIS を再起動します。

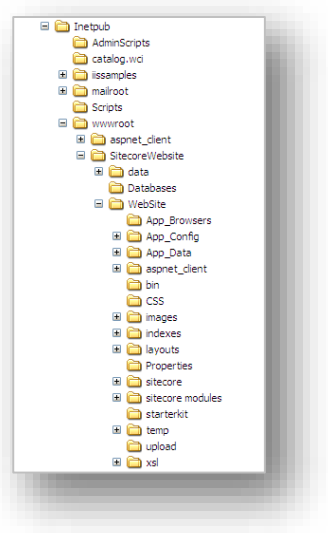
このプロセスを、`/sitecore/admin`, `/sitecore/debug` と `/sitecore/shell/WebService` フォルダーについて繰り返します。

2.4 Web サイト フォルダーの構造

次のフォルダーを Web サイトのルート フォルダーの外に配置して、セキュリティを向上させることができます。

- */data*
- */indexes*

`/data` フォルダーを移動したら、`web.config` ファイルを編集して、新しい場所を参照させる必要があります。`ASP.NET` 要求に対するアクセス許可も設定する必要があります。この詳細については、『`CMS 7.0 Installation Guide`』の「`4.2.2 File System Permissions for ASP.NET Requests`」のセクションを参照してください。



以下を使用して、Sitecore をインストールすることができます。

- インストール プログラム
- `.zip` ファイル

インストール プログラムの使用

インストール プログラムを使用して Sitecore をインストールする場合、`/data` フォルダーが Web サイトのルート フォルダーの外に作成され、`web.config` ファイルがその場所を参照するよう編集されます。`/indexes` フォルダーが `/data` フォルダー内に配置されます。

これが推奨の設定であり、変更する必要はありません。

.zip ファイルの使用

`.zip` ファイルを使用して Sitecore をインストールする場合、データ フォルダーは Web サイトのルート フォルダーの外に作成されますが、`web.config` ファイルはその場所を参照するよう編集されません。初めて Sitecore を実行するとき、`/WebSite` フォルダー内に別のデータ フォルダーが作成されます。したがって、`web.config` ファイルを正しい場所を示すように編集することを推奨します。

`web.config` ファイルは、次のようになります。

```
          <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" />
```

2.5 ログイン ページでのユーザー名のオート コンプリートの無効化

ログイン時にユーザー名の入力を自動で補完しないよう指定することにより、Sitecore インストール環境のセキュリティを向上させることもできます。

ユーザー名のオートコンプリートを無効にする方法:

1. `C:\Inetpub\wwwroot\YourWebsite\WebSite\sitecore\login` フォルダーに移動します。
2. `default.aspx` ファイルを開きます。
3. `form id="LoginForm"` セクションを探します。
4. このセクションを次のように編集します。

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

2.6 ファイルのアップロードの制御

ユーザーによってアップロードされたファイルへのアクセスを制御することにより、Sitecore インストール環境のセキュリティを強化することができます。

2.6.1 アップロード フォルダーでの実行アクセス許可の拒否

アップロード フォルダーのコンテンツをユーザーが変更できるようにすると、そのフォルダー内にスクリプトおよび実行可能プログラムを配置するアクセス許可も与えることになります。これらのスクリプトやプログラムが実行されることで、サーバーで予想外の動作が起きることがあります。したがって、アップロードされたファイルをユーザーがダウンロードしようとするときに、そのファイルがサーバー側で実行されないようにする必要があります。

`/upload` 内のスクリプトおよび実行可能ファイルの実行アクセス許可を与えないことをお勧めします。

メモ
この手順は、コンテンツ オーサーが `/upload` に直接ファイルを配置することができる設定の場合にのみ必要になります。たとえば、共有ディレクトリまたは FTP サーバーを使用する場合は、コンテンツ オーサーがメディア ライブラリに多くのメディアを即座に配置することができます。

IIS での実行アクセス許可の詳細については、http://support.microsoft.com/kb/313075 を参照してください。

一時フォルダーへのファイルのアップロードの拒否

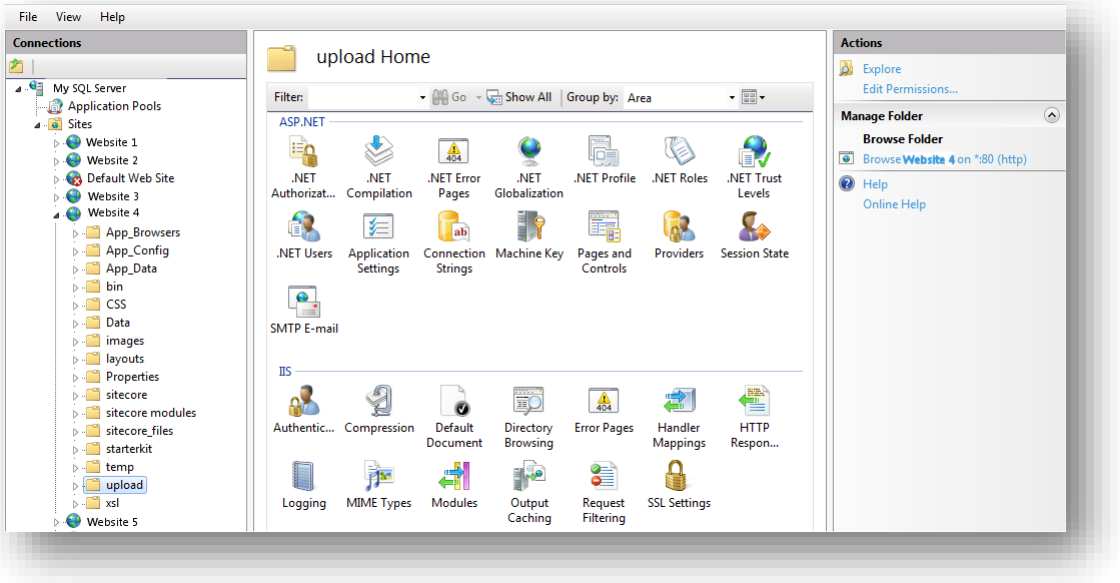
ユーザーが `/temp` フォルダーにファイルをアップロードすることも拒否してください。

メモ
この手順は、主に、コンテンツ オーサーが共有ディレクトリまたは FTP サーバーを使用して `/temp` フォルダーに直接ファイルを配置することができるように設定する場合に必要になります。ただし、.aspx ファイルが何らかの理由で (たとえばカスタム コードから) 最終的に `/temp` フォルダーに保存される場合は、潜在的なセキュリティの問題を回避するため、この手順を必ず実行することをお勧めします。

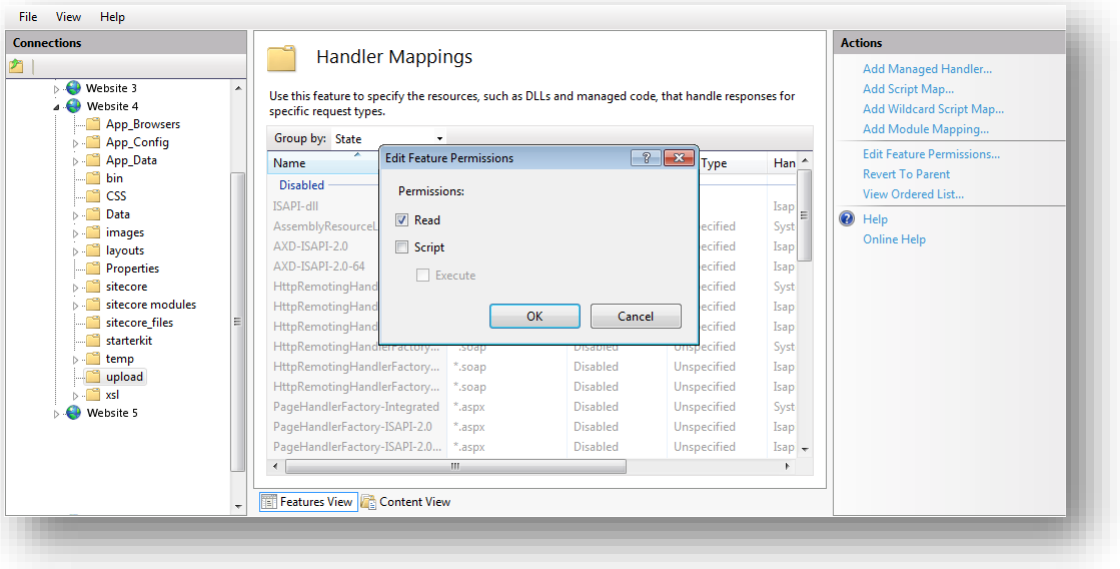
実行アクセス許可の拒否

アップロード フォルダーへのスクリプトと実行アクセス許可の両方を拒否する必要があります。

1. 対象のデータベースが存在するアップロード フォルダーに移動します。



2. アップロード フォルダーを選択し、[**ハンドラー マッピング**] をクリックして、[**操作**] ペインで [**機能のアクセス許可の編集**] をクリックします。



3. [**機能のアクセス許可の編集**] ダイアログ ボックスで、[**スクリプト**] チェック ボックスと [**実行**] チェック ボックスをオフにします

2.6.2 アップロード ウォッチャーの無効化

アップロード ウォッチャーを無効にして、**メディア ライブラリ**からのみファイルをアップロード可能にすることをお勧めします。これにより、ファイルをアップロードできるのは Sitecore クライアントからのみになり、アップロードされたファイルについて確実に制御できるようになります。

アップロード ウォッチャーを無効にすると、`/upload` フォルダーに配置されたファイルが自動的に**メディア ライブラリ**にアップロードされることはありません。

アップロード ウォッチャーを無効にするには、次の行を `Web.config` ファイルの `<modules>` セクションから削除します

```
 <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"/>
```

2.6.3 アップロード フィルター ツール

ただし、完全な制御を行い、ユーザーが特定のファイル タイプをアップロードできないようにするには、**アップロード フィルター** ツールを使用する必要があります。

アップロード フィルター ツールは、たとえば `.exe`、`.dll` など特定のファイル タイプがアップロードされないようにします。

アップロード フィルター ツール — `Upload Filter-1.0.0.2.zip` — は SDN (Sitecore Developer Network) からダウンロードできます。ここでは、セキュリティ強化ガイドおよび Sitecore パッケージ ファイルとして入手できます。

Sitecore パッケージには次のファイルが含まれます。

ファイル名	保存フォルダー
`UploadFilter.config`	`Website\App_Config\Include\`
`UploadFilter.dll`	`WebSite\bin\`

アップロード フィルター ツールのインストール

アップロード フィルター ツールを使用する前に、Sitecore パッケージ ファイルがインストールされている必要があります。

アップロード フィルター ツールをインストールする方法:

1. **Sitecore デスクトップ**で、[Sitecore]、[コントロール パネル] をクリックします。
2. **Sitecore コントロール パネル**で、[管理] をクリックし、[パッケージをインストールする] をクリックします。
3. ウィザードに従ってインストール プロセスを進めます。

アップロード フィルター ツールの設定

パッケージをインストールしたら、ツールを設定する必要があります。

アップロード フィルター ツールを設定する方法:

1. `UploadFilter.config` ファイルを開きます。

```
        <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>
```

2. `Allowed extensions` パラメーターに、アップロード可能なファイルの拡張子の種類をカンマ区切り形式で入力します。

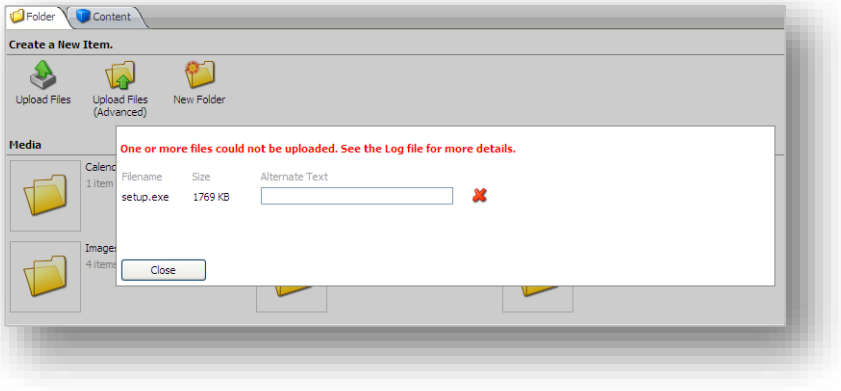
   または

   `Blocked extensions` パラメーターに、アップロードできないファイルの拡張子の種類をカンマ区切り形式で入力します。

   ファイル拡張子はドットを付けずに入力します。

   重要
   Allowed extensions パラメーターを設定すると、Blocked extensions パラメーターは無視されます

3. ブロックする拡張子リスト上にあるファイル タイプをアップロードしようとすると、次のメッセージが表示されます。



2.7 セキュリティおよびクライアント RSS フィード

RSS テクノロジは、RSS リンクをたどるユーザーが、RSS フィードの URL で指定されたアイテムに直接到達できるように設計されたものです。大半の RSS リーダーは、認証をサポートしません。つまり、Sitecore クライアントの RSS フィードに加入するユーザーは、RSS フィードの URL で指定されたアイテムに直接アクセスし、RSS フィードの参照時に Sitecore セキュリティ システムに対して身元を明らかにする必要がありません。ただし、Sitecore セキュリティ システムでは、クライアント フィードに関連付けられ任意のアクションを実行する場合には、そのユーザーが認証されていることを検証します。

RSS フィードの URL に他のユーザーがアクセスした場合、次のようになります。

- そのユーザーのセキュリティ権限ではアイテムへのアクセスが許可されていなくても、リンクをたどって、RSS フィード内に保存されているすべてのコンテンツを参照することができます。
- コンテンツ上ではどのようなアクションも実行できません
- 他のコンテンツは参照できません。
- RSS フィードの元の所有者のユーザー名やパスワードにはアクセスできません。
- 他のコンテンツへアクセスするためのリンクは修正できません。

重要
Sitecore ユーザーは RSS フィードを共有しないでください。

2.7.1 クライアント RSS フィードの無効化

Sitecore インストール環境に保護が必要な機密情報が含まれる場合、Sitecore クライアント RSS フィードを無効にすることができます。

Sitecore クライアント フィードを無効にする方法

1. `web.config` ファイルを開きます。
2. `<httpHandlers>` セクションを探します。IIS プール状況によっては、このセクションはハンドラーと呼ばれる場合もあります。
3. 次のハンドラーを削除します。

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

このハンドラーを削除すると、Sitecore 内部で使用可能なすべてのクライアント フィードが無効になります。ただし、作成したパブリック RSS フィードは、Web サイト訪問者に向けて公開されています。

2.8 応答からのヘッダーの削除

Web サイトによって送信される各応答からヘッダー情報を削除することによって、セキュリティを向上させ、帯域幅を若干節約することができます。

これらのヘッダーには、公開する必要のない Web サイトで使用されているフレームワークについてのインフラストラクチャの詳細がいくつか含まれています。

次のものは簡単に削除することができます。

- X-Aspnet-Version HTTP ヘッダー
- X-Powered-By HTTP ヘッダー
- X-AspNetMvc-Version HTTP ヘッダー

2.8.1 X-Aspnet-Version HTTP ヘッダーの削除

各 Web ページから X-Aspnet-Version HTTP ヘッダー情報を削除することによって、帯域幅が若干節約され、使用している ASP.NET のバージョンが確実に非公開になります。

ASP.NET からの各応答から、X-Aspnet-Version HTTP ヘッダーを削除するには、`web.config` file ファイルに次の行を追加します。

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

X-Aspnet-Version HTTP ヘッダーの削除の詳細については、http://www.dotnetperls.com/x-aspnet-version を参照してください。

2.8.2 X-Powered-By HTTP ヘッダーの削除

X-Powered-By HTTP ヘッダーを削除することによって、使用している ASP.NET のバージョンが非公開になります。

ASP.NET からの各応答から、X-Powered-By HTTP ヘッダーを削除するには、`web.config` file ファイルに次の行を追加します。

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

2.8.3 X-AspNetMvc-Version HTTP ヘッダーの削除

X-AspNetMvc-Version HTTP ヘッダーを削除することによって、使用している ASP.NET MVC のバージョンが非公開になります。

X-AspNetMvc-Version HTTP ヘッダーを削除するには、`Application_Start` ファイルの `Global.asax.cs` メソッドに次の行を追加します。

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

2.9 参考文献

Sitecore の安全性を向上させるための情報として、次の文書をお勧めします。

- 『Sitecore インストール ガイド』

  http://sdn.sitecore.net/Products/Sitecore%20V5/Sitecore%20CMS%207/Installation.aspx
- アップグレード前の Sitecore のバージョンからのアップグレード指示
- アップグレードする Sitecore のバージョン用の `web.config` 変更文書

2.9.1 その他の参考文献

SQL サーバーのセキュリティの詳細については、次を参照してください。

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

全般的なセキュリティの詳細については、Microsoft セキュリティ TechCenter を参照してください。

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