

Säkerhetsrisker i webbapplikationer baserade på **ASP.NET**

Christer Leuhusen

christer.leuhusen@valtech.se

Christian Westman

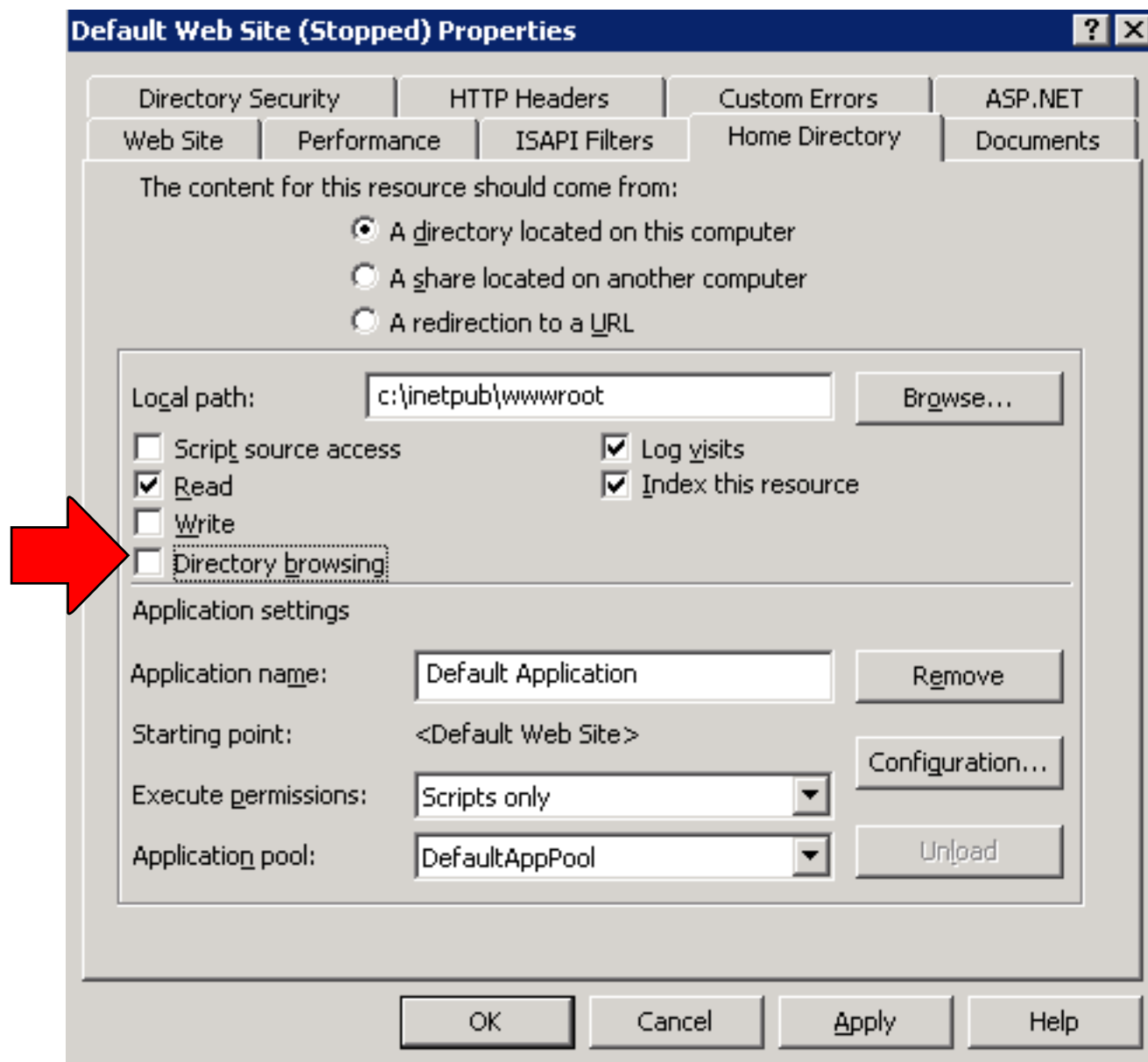
christian.westman@valtech.se

- IIS Configuration
- Website Configuration
- Log Files
- Input Validation
- Output Control
- SQL Injection
- File Upload
- Protecting Web.Config

IIS Configuration

IIS Configuration

Disable directory browsing



IIS Configuration

Run anonymous access with low permissions account

The image shows two overlapping windows from the IIS configuration console. The 'Util Properties' window is on the left, and the 'Authentication Methods' window is on the right. Two red arrows point from the 'Authentication and access control' section of the 'Util Properties' window to the 'Authentication Methods' window.

Util Properties

Directory Security | HTTP Headers | Custom Errors | ASP.NET

Authentication and access control

Enable anonymous access and edit the authentication methods for this resource. [Edit...](#)

IP address and domain name restrictions

Grant or deny access to this resource using IP addresses or Internet domain names. [Edit...](#)

Secure communications

Require secure communications and enable client certificates when this resource is accessed. [Server Certificate...](#) [View Certificate...](#) [Edit...](#)

OK Cancel Apply Help

Authentication Methods

Enable anonymous access

Use the following Windows user account for anonymous access:

User name: [Browse...](#)

Password:

Authenticated access

For the following authentication methods, user name and password are required when:

- anonymous access is disabled, or
- access is restricted using NTFS access control lists

Integrated Windows authentication

Digest authentication for Windows domain servers

Basic authentication (password is sent in clear text)

.NET Passport authentication

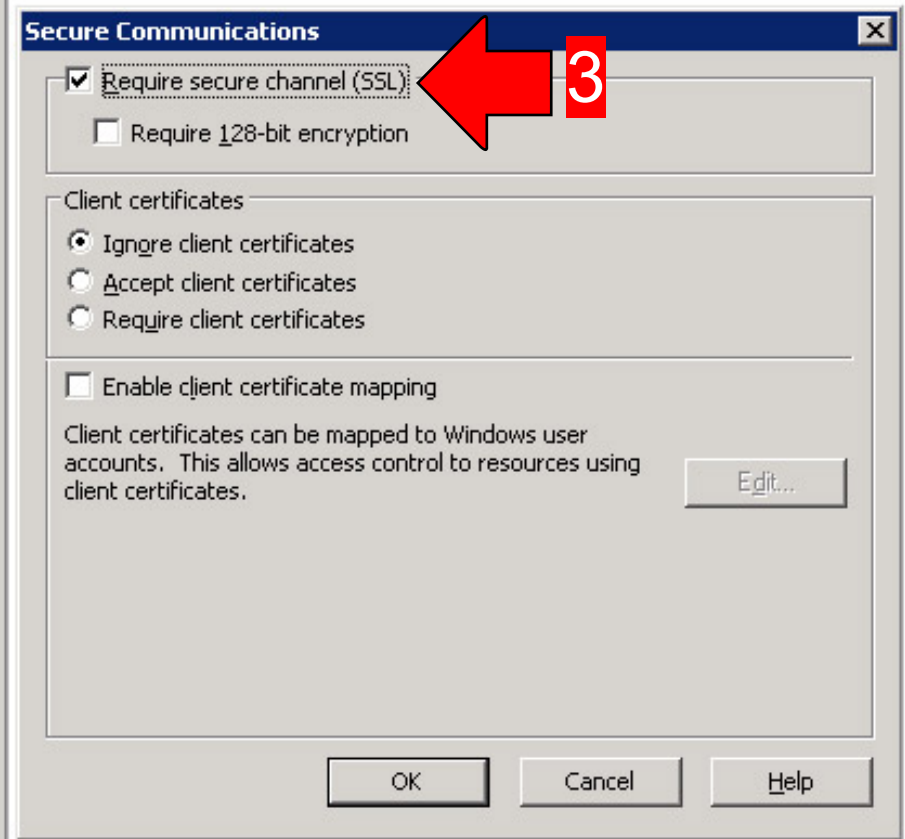
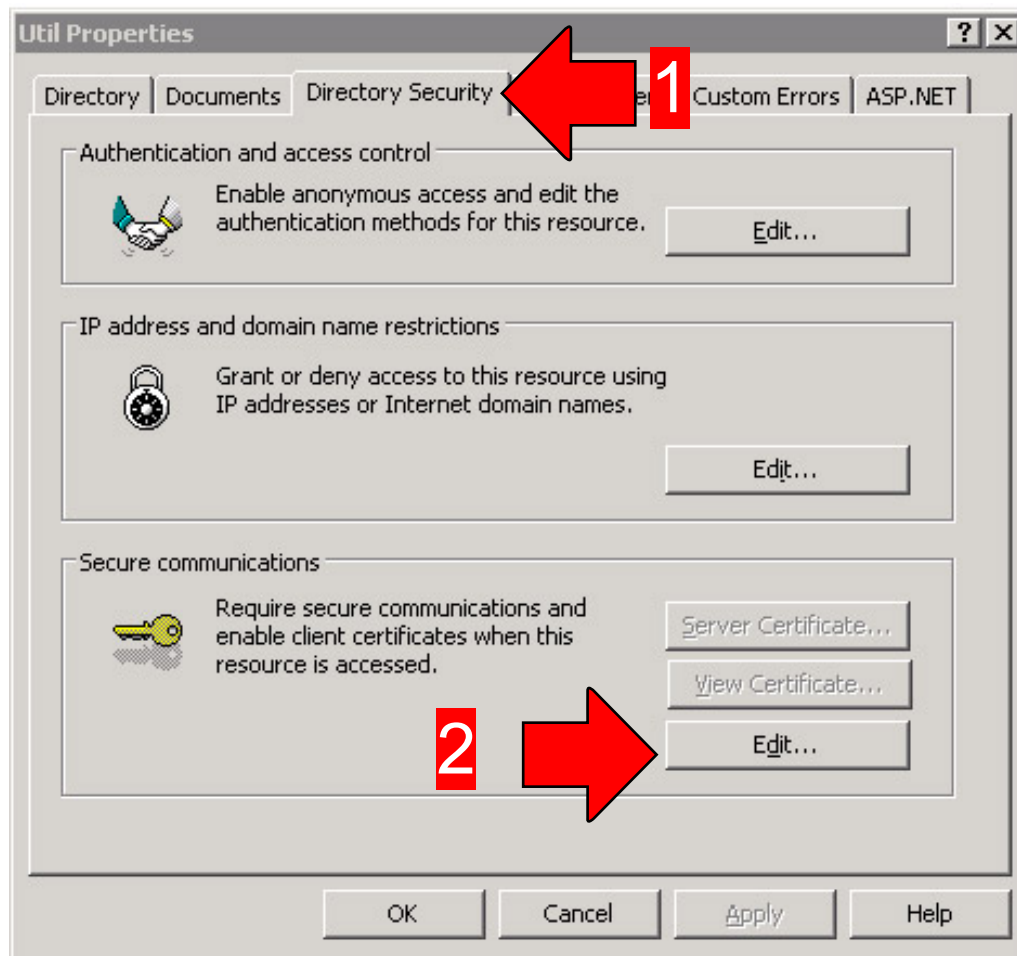
Default domain: [Select...](#)

Realm: [Select...](#)

OK Cancel Help

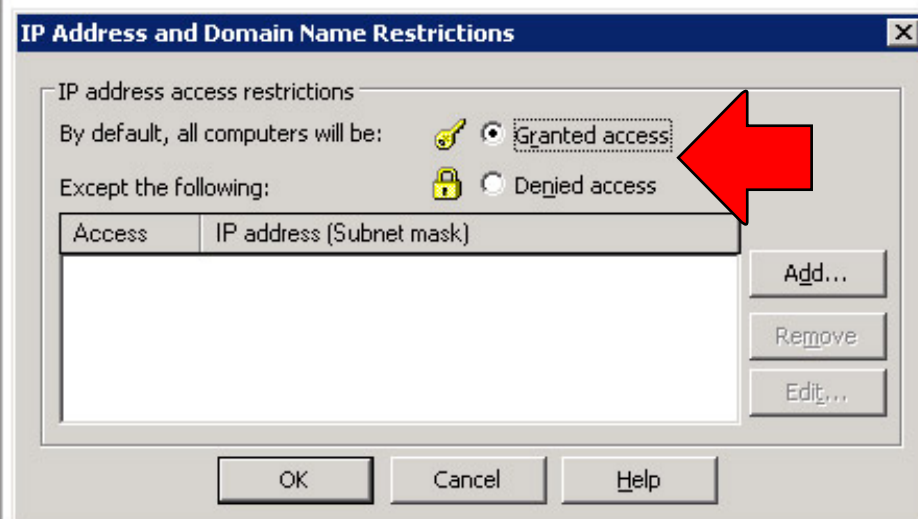
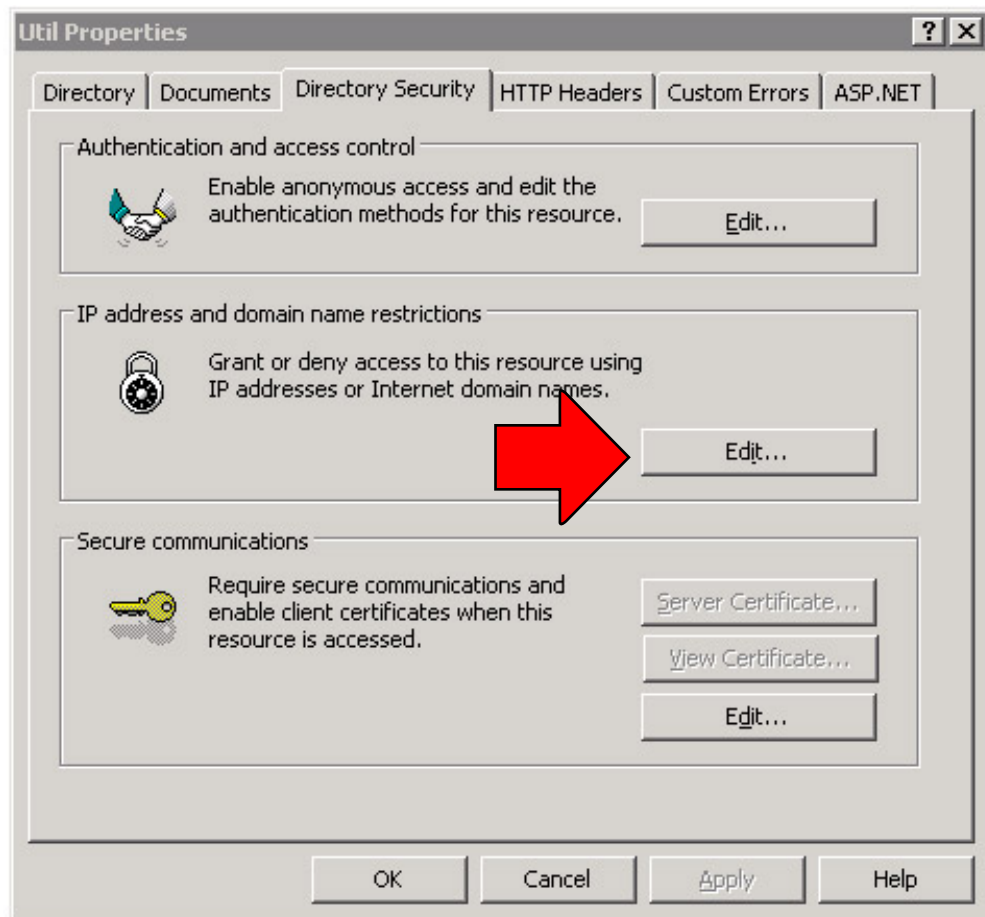
IIS Configuration

Enable SSL on folders containing edit/admin-mode
(Right-click folder to protect)



IIS Configuration

Restrict edit/admin-folders to valid IP:s only



Website Configuration

Website Configuration

Do you need ViewState?

```
<pages
```

```
...
```

```
  enableViewState="true, false"
```

```
...
```

```
/>
```

- *Disable in web.config and enable where needed*

Website Configuration

Do you need to tamper proof ViewState?

```
<pages
```

```
...
```

```
enableViewStateMac="true, false"
```

```
...
```

```
/>
```

- *provides viewstate integrity*
- *some impact on performance*

Website Configuration

Do you need to encrypt ViewState?

```
.  
<pages
```

```
...
```

```
  viewStateEncryptionMode="Auto, Always, Never"
```

```
...
```

```
/>
```

- *provides viewstate confidentiality & integrity*
- *some impact on performance*

Website Configuration

How should you set up the encryption?

```
<machineKey  
  validationKey="AutoGenerate,IsolateApps"  
  decryptionKey="AutoGenerate,IsolateApps"  
  validation="AES, DES, 3DES"  
  decryption="AES, DES, 3DES"  
>
```

- *Autogenerate for non webfarms*
- *Use your own/same key on all webfarm servers*
- *Preferably AES over DES or 3DES*

Website Configuration

Keeping the bulk of the trash out

```
<pages
```

```
...
```

```
  validateRequest="true, false"
```

```
...
```

```
/>
```

- *prevents posting of non escaped tagged content*
- *should be set to **true** in web.config*
- *override when needed in asp page headers*
- *not enabling will, among other things, lead to **XSS** attacks*

Website Configuration

Disable dynamic compilation

```
<pages
```

```
...
```

```
  compilationMode="Always, Auto, Never"
```

```
...
```

```
/>
```

- *consider setting to **Never** to discourage execution of unwanted code.*
- *Setting this to Never requires the web application to be precompiled.*
- *can be overridden*

Server Error in '/' Application.

Exception of type 'System.Exception' was thrown.

Description: An unhandled exception occurred during the execution of the current web request. Please review the stack trace for more information about the error and where

Exception Details: System.Exception: Exception of type 'System.Exception' was thrown.

Source Error:

```
Line 20:         {  
Line 21:             var credentials = new NetworkCredential("admin", "secretpassword");  
Line 22:             throw new System.Exception();  
Line 23:         }  
Line 24:     }
```

Source File: C:\Projects\SecurityDemo\SecurityDemo\Exception\default.aspx.cs **Line:** 22

Stack Trace:

```
[Exception: Exception of type 'System.Exception' was thrown.]  
SecurityDemo.Exception._default.Page_Load(Object sender, EventArgs e) in C:\Projects\SecurityDemo\SecurityDemo\Exception\default.aspx.cs:22  
System.Web.Util.CalliHelper.EventArgFunctionCaller(IntPtr fp, Object o, Object t, EventArgs e) +14  
System.Web.Util.CalliEventHandlerDelegateProxy.Callback(Object sender, EventArgs e) +35  
System.Web.UI.Control.OnLoad(EventArgs e) +99  
System.Web.UI.Control.LoadRecursive() +50  
System.Web.UI.Page.ProcessRequestMain(Boolean includeStagesBeforeAsyncPoint, Boolean includeStagesAfterAs
```

Version Information: Microsoft .NET Framework Version:2.0.50727.4927; ASP.NET Version:2.0.50727.4927

Website Configuration

Fail? Do not give out the details

```
<customErrors mode="Off, On, RemoteOnly" defaultRedirect="ErrorPages/oops.html">
```

```
<error statusCode="500" redirect="ErrorPages/500.html" />
```

```
</customErrors>
```

- Should be set to **On** or at least **RemoteOnly** on publicly exposed servers.
- Should preferably redirect to **static** html pages
- Make sure to have a **defaultRedirect** set

Log files

Log files

- Do not store in website directory
- Do not store on the system partition
- Preferably stored on a separate drive
- Most secure: store to CD-drive that's writable only

Input Validation

Input Validation

ASP.NET *ValidationControls*

```
<form id="form1" runat="server">
```

```
<asp:TextBox runat="server" ID="Name" />
```

```
<asp:Button runat="server" Text="Submit your name" />
```

```
</form>
```

Input Validation

ASP.NET *ValidationControls*

```
.  
<form id="form1" runat="server">
```

```
.  
<asp:RegularExpressionValidator runat="server" ID="NameValidator" ControlToValidate="Name" ValidationExpression="^[a-öA-Ö\s]{2,50}$" Text="this does not look like a name to me" EnableClientScript="true"/>
```

```
<asp:TextBox runat="server" ID="Name" />
```

```
<asp:Button runat="server" Text="Submit your name" />  
</form>
```

Input Validation

ASP.NET ValidationControls

.
. .
.

```
if (IsValid)  
    SaveFormData();
```

Input Validation

Web Services

- validate all web service input
- never trust external input

SQL injection

SQL injection

Concatenated strings = BAD!

```
Email.Text = "' or 'true'='true'--";  
Password.Text = "foo!";
```

```
var sql = @"select * from subscribers where  
            email = " + Email.Text +  
            "and password = " + Password.Text + """;
```

Resulting SQL:

```
select * from subscribers where email="' or 'true' = 'true'-- and  
password = 'foo!'
```

SQL injection

Parameters = GOOD

```
var sql = @"select * from subscribers where  
email = @email and password = @password";
```

```
sql.Parameters.Add(new SqlParameter("email", DbType.String));  
sql.Parameters["email"].Value = Email.Text;
```

```
sql.Parameters.Add(new SqlParameter("password", DbType.String));  
sql.Parameters["password"].Value = Password.Text;
```

SQL injection will not succeed, phew

File Upload

File Upload

verify that the file extension is what you expect

```
var validExtensions = new string[3] { ".gif", ".jpg", ".png" };
```

```
var filename = FileUpload.FileName;
```

```
var fileExtension = filename.Substring(filename.LastIndexOf('.')).ToLower();
```

```
if(!validExtensions.Contains(fileExtension))
```

```
    throw new InvalidDataException("no thank you");
```

```
else
```

```
    but is it really an image?
```

File Upload

But is it really an image?

```
try
{
    using (var im = Image.FromStream(FileUpload.FileContent))
    {
        //seems to be an image, process/save it or something
    }
}
catch (ArgumentException ex)
{
    //Maybe someone cares enough to check the logs
    Log.Warn("Someone attempted to upload something nasty", ex);
    throw;
}
```

File Upload

You are in control

- Avoid saving uploaded files using the original filename. Use some randomness in their names, GUIDs perhaps
- Do not store uploaded files as so they are directly accessible from the web unless you are dead sure they are safe.
- Never store uploaded files on the system partition or you will get DOS:ed
- If possible, store uploaded files on a different drive separated from the web drive.

Output Control

Output Control

Comment out - really comment out - or Delete?

```
<form runat="server" runat="server">
```

//this will render to browser

```
<!--<input type="hidden" name="admin" value="123">-->
```

```
<input type="text" name="username"/>
```

```
<input type="password" name="password"/>
```

```
<input type="submit" value="Login"/>
```

```
</form>
```

Output Control

Comment out - really comment out - or Delete?

```
<form runat="server" runat="server">
```

//this will not render to the browser

```
<%--<input type="hidden" name="admin" value="123">--%>
```

```
<input type="text" name="username"/>
```

```
<input type="password" name="password"/>
```

```
<input type="submit" value="Login"/>
```

```
</form>
```

Output Control

Comment out - really comment out - or Delete?

```
<form runat="server" runat="server">
```

```
//this will not render to the browser  
..deleted..
```

```
<input type="text" name="username"/>
```

```
<input type="password" name="password"/>
```

```
<input type="submit" value="Login"/>
```

```
</form>
```

Web.Config Encryption

Web.Config Encryption

```
<connectionStrings>  
  <add  
    name="ApplicationDatabase"  
    connectionString="Data Source=127.0.0.14;  
      Initial Catalog=AppDB;  
      Integrated Security=False;  
      User ID=appuser;  
      Password=@tl4.s0;  
      Connect Timeout=10"  
    providerName="System.Data.SqlClient" />  
</connectionStrings>
```

Web.Config Encryption

`aspnet_regiis.exe - encrypt`

```
aspnet_regiis.exe -pef "connectionStrings" E:  
\Websites\MyWebSite -prov  
"DataProtectionConfigurationProvider"
```

- `%WinDir%\Microsoft.NET\Framework\<versionNumber>`

Web.Config Encryption

```
<connectionStrings configProtectionProvider="
DataProtectionConfigurationProvider">
<EncryptedData>
<CipherData><CipherValue>AQAAANCMnd8BFdERjHoAwE/
Cl+sBAAAaexuIJ/8oFE+sGTs7jBkZdgQAAAACAAAAAADZ
gAAqAAAABAAAAA
Kms84dyaCPAeaSC1dIMIBAAAAAASAAACgAAAAEAAAAKa
VI6aAOFdqhdc6w1Er3HMwAAAACZ00MZOz1dI7kYRvkMIn/
BmfrvoHNUwz6H9rcxJ6Ow41E3hwHLbh79IUWiiNp0VqFAA
AAF2sXCdb3fcKkgnagkHkILqteTXh</CipherValue>
</CipherData>
</EncryptedData></connectionStrings>
```

Web.Config Encryption

aspnet_regiis.exe - **decrypt**

aspnet_regiis.exe -pdf "connectionStrings" E:\Websites\MyWebSite

Thank You!