Code Access Security in SharePoint 2007 for Administrators

By Brett Lonsdale, MCSD.NET , MCT
Combined Knowledge
www.combined-knowledge.com
brett@combined-knowledge.com

This 'How to' guide will take you through configuring Code Access Security for SharePoint. The reason for this white paper is to provide a basic understanding of Code Access Security to Administrators of SharePoint. This will enable Administrators to deploy Web Parts to SharePoint safely without unnecessarily upping the trust level for the entire Web Application.

My server topology for this guide was as follows:

Server Name	**Role**	**Products Installed**
Rootdc	**Domain Controller**	**Windows server 2003 SP1**
SQL	**Database server**	**SQL 2000 SP4**
MOSS2007	**MOSS2007 Server**	**MOSS2007 Enterprise**

Code Access Security – Primer

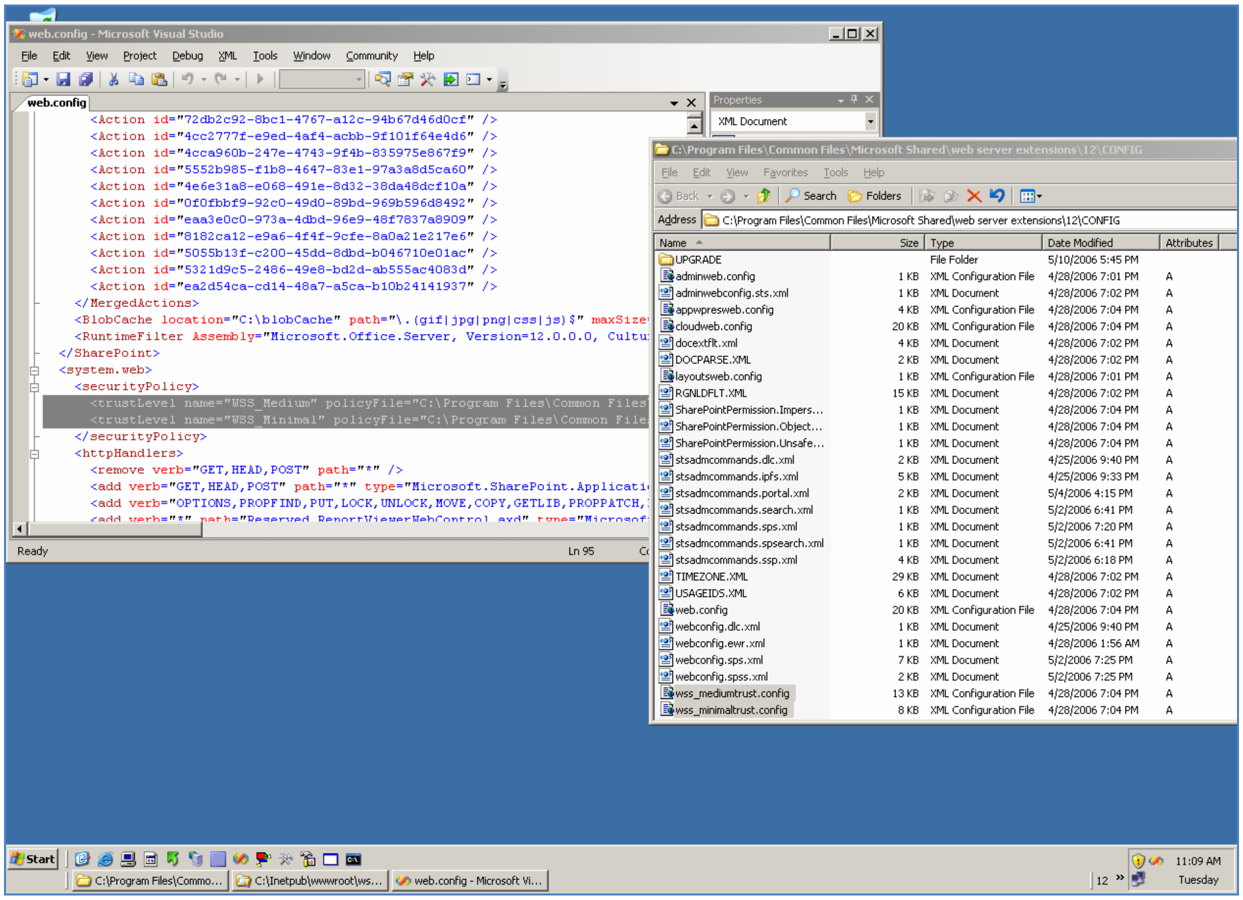
Historically Administrators have been concerned primarily with Security for Users and not with Security for Code. The .NET framework has also provided us with Code Access Security so that we can apply permissions to Code as well as users. SharePoint Web Parts run with a Trust Level which is set in the Web.Config file for the entire Web Application allowing not just one Web Part, but all Web Parts to run with that trust. Many developers request Administrators to set the Trust Level for the Web Application to 'Full' so that their Web Part can carry out the actions it was designed to perform. However, by doing that, you are allowing all Web Parts to run with Full Trust. This is the equivalent of saying 'I need an Administrator of our Network, therefore I will give everyone Administrator permissions'.

The default trust level that the Web Application runs with is WSS_Minimal. Developers will normally request that this is set to WSS_Medium or Full which are the only other configurations possible unless the developer creates a custom configuration file.

The intent of this white paper is to descibe the risk of changing the Trust Level and also so that you have the background knowledge of Code Access Security so that you can ensure Developers have done their jobs properly.

Code Access Security – The Architecture

SharePoint 2007 provides two configuration files for two different trust levels: WSS_Minimal and WSS_Medium. A third configuration 'Full' can also be set which uses one of the ASP.NET trust levels. In the Web.Config file for the web application you will see two Trust Level Elements in the Security Policy Element pointing to the two configuration files stored in 12\CONFIG (localdrive:\program files\common files\microsoft shared\web server extensions\12).


web.config - Microsoft Visual Studio
File Edit View Project Debug XML Tools Window Community Help
web.config
Properties
XML Document
Ready
Ln 95
File Edit View Favorites Tools Help
Back Search Folders
Address C:\Program Files\Common Files\Microsoft Shared\web server extensions\12\CONFIG
Start
C:\Program Files\Commo...
C:\Inetpub\wwwroot\ws...
web.config - Microsoft Vi...
11:09 AM
Tuesday

```
        <Action id="72db2c92-8bc1-4767-a12c-94b67d46d0cf" />
        <Action id="4cc2777f-e9ed-4af4-acbb-9f101f64e4d6" />
        <Action id="4cca960b-247e-4743-9f4b-835975e867f9" />
        <Action id="5552b985-f1b8-4647-83e1-97a3a8d5ca60" />
        <Action id="4e6e31a8-e068-491e-8d32-38da48dcf10a" />
        <Action id="0f0fbbf9-92c0-49d0-89bd-969b596d8492" />
        <Action id="eaa3e0c0-973a-4dbd-96e9-48f7837a8909" />
        <Action id="8182ca12-e9a6-4f4f-9cfe-8a0a21e217e6" />
        <Action id="5055b13f-c200-45dd-8dbd-b046710e01ac" />
        <Action id="5321d9c5-2486-49e8-bd2d-ab555ac4083d" />
        <Action id="ea2d54ca-cd14-48a7-a5ca-b10b24141937" />
      </MergedActions>
      <BlobCache location="C:\blobCache" path="\.(gif|jpg|png|css|js)$" maxSize
      <RuntimeFilter Assembly="Microsoft.Office.Server, Version=12.0.0.0, Cultu
    </SharePoint>
    <system.web>
      <securityPolicy>
        <trustLevel name="WSS_Medium" policyFile="C:\Program Files\Common Files
        <trustLevel name="WSS_Minimal" policyFile="C:\Program Files\Common File
      </securityPolicy>
      <httpHandlers>
        <remove verb="GET,HEAD,POST" path="*" />
        <add verb="GET,HEAD,POST" path="*" type="Microsoft.SharePoint.Applicati
        <add verb="OPTIONS,PROPFIND,PUT,LOCK,UNLOCK,MOVE,COPY,GETLIB,PROPPATCH,
        <add verb="*" path="Reserved.ReportViewerWebControl.axd" type="Microsof
```

Name	Size	Type	Date Modified	Attributes
UPGRADE		File Folder	5/10/2006 5:45 PM	
adminweb.config	1 KB	XML Configuration File	4/28/2006 7:01 PM	A
adminwebconfig.sts.xml	1 KB	XML Document	4/28/2006 7:02 PM	A
appwpresweb.config	4 KB	XML Configuration File	4/28/2006 7:04 PM	A
cloudweb.config	20 KB	XML Configuration File	4/28/2006 7:04 PM	A
docextflt.xml	4 KB	XML Document	4/28/2006 7:02 PM	A
DOCPARSE.XML	2 KB	XML Document	4/28/2006 7:02 PM	A
layoutsweb.config	1 KB	XML Configuration File	4/28/2006 7:01 PM	A
RGNLDFLT.XML	15 KB	XML Document	4/28/2006 7:02 PM	A
SharePointPermission.Impers...	1 KB	XML Document	4/28/2006 7:04 PM	A
SharePointPermission.Object...	1 KB	XML Document	4/28/2006 7:04 PM	A
SharePointPermission.Unsafe...	1 KB	XML Document	4/28/2006 7:04 PM	A
stsadmcommands.dlc.xml	2 KB	XML Document	4/25/2006 9:40 PM	A
stsadmcommands.ipfs.xml	5 KB	XML Document	4/25/2006 9:33 PM	A
stsadmcommands.portal.xml	2 KB	XML Document	5/4/2006 4:15 PM	A
stsadmcommands.search.xml	1 KB	XML Document	5/2/2006 6:41 PM	A
stsadmcommands.sps.xml	1 KB	XML Document	5/2/2006 7:20 PM	A
stsadmcommands.spsearch.xml	1 KB	XML Document	5/2/2006 6:41 PM	A
stsadmcommands.ssp.xml	4 KB	XML Document	5/2/2006 6:18 PM	A
TIMEZONE.XML	29 KB	XML Document	4/28/2006 7:02 PM	A
USAGEIDS.XML	6 KB	XML Document	4/28/2006 7:02 PM	A
web.config	20 KB	XML Configuration File	4/28/2006 7:04 PM	A
webconfig.dlc.xml	1 KB	XML Document	4/25/2006 9:40 PM	A
webconfig.ewr.xml	1 KB	XML Document	4/28/2006 1:56 AM	A
webconfig.sps.xml	7 KB	XML Document	5/2/2006 7:25 PM	A
webconfig.spss.xml	2 KB	XML Document	5/2/2006 7:25 PM	A
wss_mediumtrust.config	13 KB	XML Configuration File	4/28/2006 7:04 PM	A
wss_minimaltrust.config	8 KB	XML Configuration File	4/28/2006 7:04 PM	A

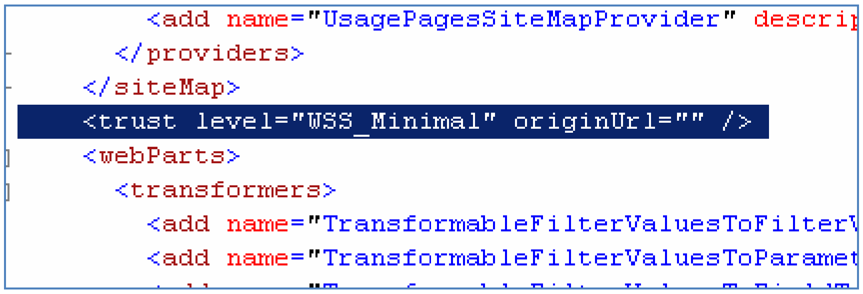
Each configuration file configures a number of classes by setting its properties. The classes are illustrated below. If we take EnvironmentPermission for example WSS_Medium uses EnvironmentPermission and sets its Read property allow Reading of "TEMP;TMP;USERNAME;OS;COMPUTERNAME". Another example is the WebPartPermission which has a Connections Property set to True for WSS_Minimal. Without this permission, your Web Parts would be unable to connect to each other.

The Table Below describes the Permission classes that are used and configured in the two configuration files.

Permission	Description	WSS_Minimal	WSS_Medium
AllMemberShipCondition	Represents a membership condition that matches all code.	*	*
AspNetHostingPermission	Controls access permissions in ASP.NET hosted environments.	*	*
DnsPermission	Controls rights to access Domain Name System (DNS) servers on the network.		*
EnvironmentPermission	Controls access to system and user environment variables.		*
FileIOPermission	Controls the ability to access files and folders.		*
FirstMatchCodeGroup	Allows security policy to be defined by the union of the policy statement of a code group and that of the first child code group that matches.	*	*
IsolatedStorageFilePermission	Specifies the allowed usage of a private virtual file system.		*
NamedPermissionSet	Defines a permission set that has a name and description associated with it.	*	*
PrintingPermission	Controls access to printers.		*
SecurityPermission	Describes a set of security permissions applied to code.	*	*
SharePointPermission	The **SharePointPermission** class represents a custom permission that controls the ability to access Microsoft SharePoint Products and Technologies resources.		*
SmtpPermission	Controls access to Simple Mail Transport Protocol (SMTP) servers.		*
SqlClientPermission	Enables the .NET Framework Data Provider for SQL Server to help make sure that a user has a security level sufficient to access a data source.		*
StrongNameMembershipCondition	Determines whether an assembly belongs to a code group by testing its strong name.	*	*
UIPermission	Controls the permissions related to user interfaces and the clipboard.		*
UnionCodeGroup	Represents a code group whose policy statement is the union of the current code group's policy statement and the policy statement of all its matching child code groups.	*	*
UrlMembershipCondition	Determines whether an assembly belongs to a code group by testing its URL.	*	*
WebPermission	Controls rights to access HTTP Internet resources.		*

WebPartPermission	Represents a custom permission that controls the ability to access Web Part resources.	*	*
ZoneMembershipCondition	Determines whether an assembly belongs to a code group by testing its zone of origin.	*	*

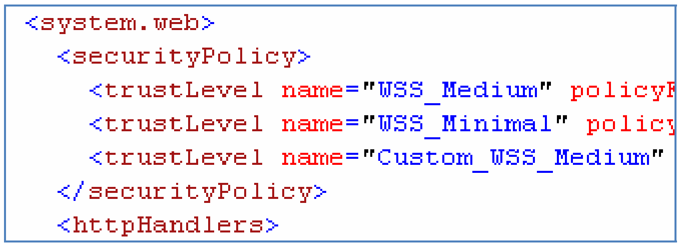
The Configuration file that you wish to use for the Web Application is set further down the web.config file in the Trust Element.



```
        <add name="UsagePagesSiteMapProvider" descri
      </providers>
    </siteMap>
    <trust level="WSS_Minimal" originUrl="" />
    <webParts>
      <transformers>
        <add name="TransformableFilterValuesToFilter
        <add name="TransformableFilterValuesToParamet
```

The default Trust is WSS_Minimal which allows most code to run but would block your code from accessing SQL, The SharePoint Object Model, Environment Variables such as Machine Name and DNS etc. To allow these permission you would need to up the trust level to WSS_Medium, Full or to a Custom Policy. However, increasing the Trust Level for the Web Application would allow all Web Parts within that Web Application to run with more permissions. Best practice would be to create a custom policy and listing your web part in the file to run with the permissions it needs.

To create a new set of permissions you need to create a new file such as Custom_WSS_Medium.config and then make SharePoint aware of the file by pointing to it in the Web.Config for the Web Application. You can merge the settings in WSS_Minimal and WSS_Medium in your new Custom File to create your own permission set. It is advised not to change the original files as Microsoft may alter these files during the application of a Service Pack. See below image:



```
<system.web>
  <securityPolicy>
    <trustLevel name="WSS_Medium" policyF
    <trustLevel name="WSS_Minimal" policy
    <trustLevel name="Custom_WSS_Medium"
  </securityPolicy>
  <httpHandlers>
```

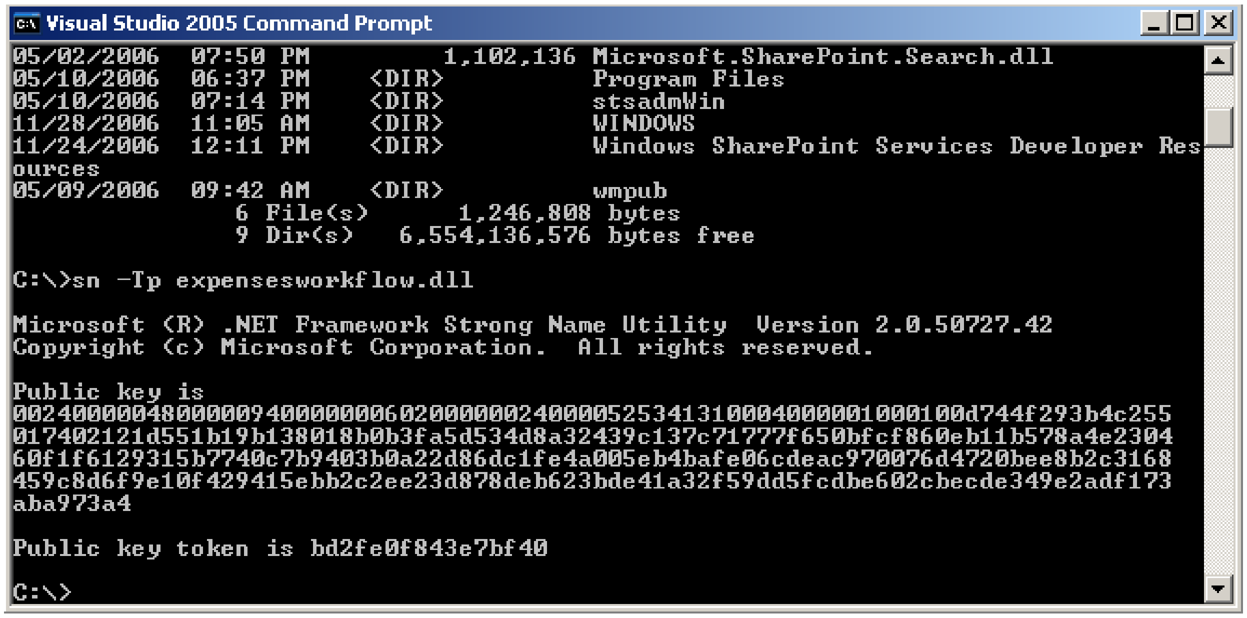
In each file WSS_Minimal.config & WSS_Medium.config there is an entry for Microsoft Web Parts allowing them to run with Full permissions.   This means that any third party Web Part is restricted by the permission classes but any Web Part with the Microsoft strong name receives Full permissions.

See below example:

```xml
<CodeGroup
        class="UnionCodeGroup"
        version="1"
        PermissionSetName="FullTrust"
        Name="Microsoft_Strong_Name"
        Description="This code group grants code signed with the
    <IMembershipCondition
            class="StrongNameMembershipCondition"
            version="1"
            PublicKeyBlob="00240000048000009400000006020000002400
    />
</CodeGroup>
```

Each custom Web Part should be Strongly Named by the Developer which provides the DLL file for the Web Part with a Public Key. You can extract the Public Key using the SN tool provided by Visual Studio.NET: sn –Tp filename.dll

This will return the Public Key BLOB and the Public Key Token for the dll. Normally this would be a developer role as it requires Visual Studio.NET.



```
05/02/2006  07:50 PM        1,102,136 Microsoft.SharePoint.Search.dll
05/10/2006  06:37 PM    <DIR>          Program Files
05/10/2006  07:14 PM    <DIR>          stsadmWin
11/28/2006  11:05 AM    <DIR>          WINDOWS
11/24/2006  12:11 PM    <DIR>          Windows SharePoint Services Developer Res
ources
05/09/2006  09:42 AM    <DIR>          wmpub
               6 File(s)      1,246,808 bytes
               9 Dir(s)   6,554,136,576 bytes free

C:\>sn -Tp expensesworkflow.dll

Microsoft (R) .NET Framework Strong Name Utility  Version 2.0.50727.42
Copyright (c) Microsoft Corporation.  All rights reserved.

Public key is
0024000004800000940000000602000000240000525341310004000001000100d744f293b4c255
017402121d551b19b138018b0b3fa5d534d8a32439c137c71777f650bfcf860eb11b578a4e2304
60f1f6129315b7740c7b9403b0a22d86dc1fe4a005eb4bafe06cdeac970076d4720bee8b2c3168
459c8d6f9e10f429415ebb2c2ee23d878deb623bde41a32f59dd5fcdbe602cbecde349e2adf173
aba973a4

Public key token is bd2fe0f843e7bf40

C:\>
```

The Public Key Token is used in the Safe Controls entry in the Web.Config for the Web Part (Assembly) to register it as a Safe Control within the Web Application. The Public Key can also be used in the custom Trust Level Configuration File to give one particular Web Part or a group of Web Parts Full Trust whilst still leaving the Trust Level set to a minimal set of permissions.

To list a Web Part with Full Permissions within your Web Application whilst still retaining a WSS_Minimal permission set for all other Web Parts follow these steps:

1. Make a copy of the WSS_Minimal.Config file from the 12\Config folder and paste it into the same folder renaming it to Custom_WSS_Minimal.Config.
2. Edit the Custom_WSS_Minimal.Config file with an XML editor such as Visual Studio.NET or NotePad.
3. Obtain the Public Key Token and Public Key Blob for the Web Part assembly that you want to deploy using the following command: sn –Tp filename.dll.
4. Create a new entry in your Custom_WSS_Minimal.Config file for your Web Part:

```
<CodeGroup
  class="UnionCodeGroup"
  version="1"
  PermissionSetName="FullTrust"
  Name="My custom Strong name"
  Description="This code group grants code signed with the “My Custom
       Strong Name” strong name full trust.">
   <IMembershipCondition class="StrongNameMembershipCondition"
   version="1"
PublicKeyBlob="0024000004800000940000000602000000240000525341310004000001
0001009BC5B83BBD16C6ABB44BAB156CF9C55D1D67078CE98CE7B423C72EE91647BD793241B
FC700192ED32481CA7CF06C205BD4C7BEDA3D4FA5ED5689AA23BC1C2118EBFA6018238AD5B
4DA420FB335E6BECA9EDE5E3F53C4BFFF266411445AE7B1D4DD14FB0F01C075589770EABD4
B1D1A70F731C098F22EDE9838CC9783C73451B5"/>
</CodeGroup>
```

5. Create a new TrustLevel element for your config file in the Web.Config called Custom_WSS_Minimal that points to your custom file in the 12\config folder.
6. Set the Trust Level of the Web.Config to Custom_WSS_Minimal
7. Recycle the Application Pool or run IISReset
8. Deploy & Test your Web Part.

I hope you found this White Paper useful, I would be grateful for any comments you may have and whether it was useful or not.

Brett Lonsdale

brett@combined-knowledge.com