Help

HeaderBlock

User Guide

Author

mark.woan@contextis.co.uk

Introduction

With the increased popularity of the Microsoft IIS7 web server, it is important that specific security recommendations can be applied to the latest web server technologies. Whilst minimising the attack surface is not a panacea it is a step towards improving security.

With the introduction of new Microsoft frameworks such as ASP.Net and MVC it appears that the number of HTTP headers returned by the IIS web server is increasing. An example of these headers is shown below:

```
Server: Microsoft-IIS/7.5
X-AspNetMvc-Version: 2.0
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
```

The commonly recommended method for removing the headers involves a combination of URLScan, application web.config changes and changes via the IIS management user interface. However, this is not convenient for large scale infrastructures and it should also be noted that the Server header cannot be removed by any of these methods for IIS 7.

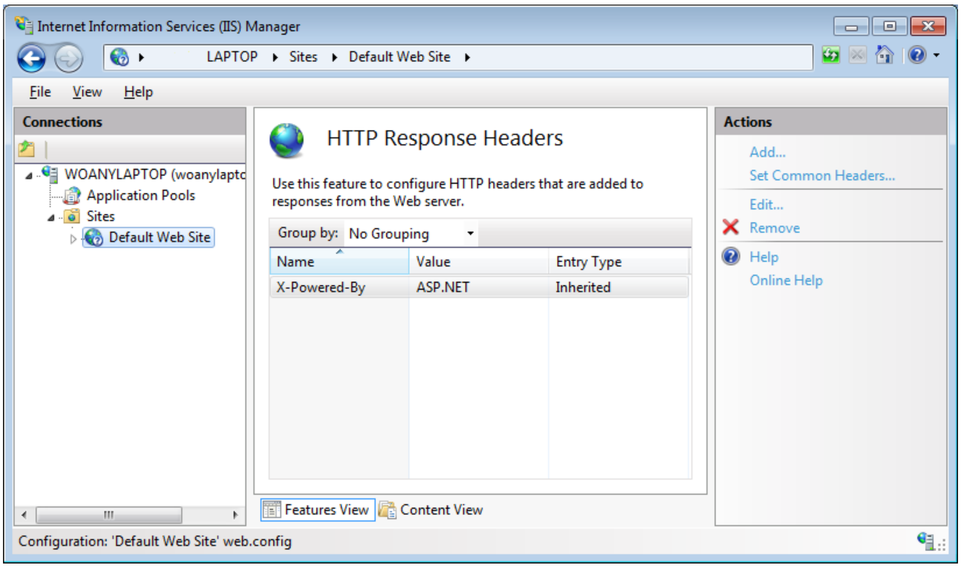
The ASP.NET pipeline allows HTTP modules to plug in into the request processing lifecycle and perform work at various stages. For example, output caching, authentication, authorization etc. are all implemented as HTTP modules. Each module must implement the IHttpModule interface. The HeaderBlock module implements the IHttpModule interface to allow the modification of HTTP responses.

Information

The HttpBlock module is written using the Microsoft C# language. The actual implementation is simple; it uses an internal list to maintain a number of HTTP headers. Before each response is sent from IIS, the module receives the response; and determines if any of the listed headers are contained within the response. If any of the headers are identified then they are removed from the response. The current list of blocked headers is as follows:

- Server
- X-AspNet-Version
- X-AspNetMvc-Version

It should be noted that the "X-Powered-By" header cannot be removed by this method. It must be removed using the IIS Manager application:



Use this feature to configure HTTP headers that are added to responses from the Web server.

Group by: No Grouping

Name	Value	Entry Type
X-Powered-By	ASP.NET	Inherited

Configuration: 'Default Web Site' web.config

IHttpModules are required to be installed into the Global Assembly Cache (GAC). Each module that is installed into the GAC must be signed. The module is supplied as source code so that users can modify the module as required. Alternatively an installer can be used. Note that regardless of the initial installation method, at least one configuration file must be altered before the header blocking functionality will work.

Instructions

The module can be used either by compiling the source code and using the generated binaries or by using supplied installer. Both methods require the editing of a configuration file to reference the binary module.

If you want to compile the module yourself then go to the **Source** section, follow the instructions there and then move onto the **Configuration** section.

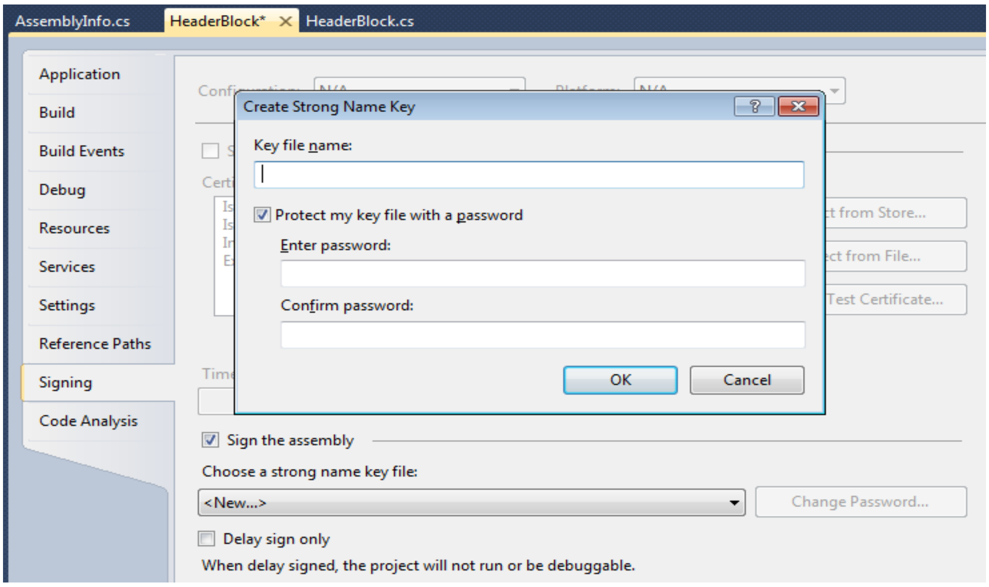
If you are using the installer then run the supplied installer and move onto the **Configuration** section. The installer has been created with Inno Setup and will install the assembly into the GAC. The installer can be found in the Installer directory.

Source

When compiling the module a strong name file must be supplied. A strong name file is used to allow multiple versions of a component to reside on a system. The strong name mechanism also ensures that the binary is digitally signed, which is a requirement of all files that are installed into the GAC.

The following steps demonstrate the actions required to compile the module:

1. Create a new strong name file. The screenshot below shows the dialog used to sign an assembly:



Key file name:

☑ Protect my key file with a password

Enter password:

Confirm password:

☑ Sign the assembly

Choose a strong name key file: <New...>

□ Delay sign only

When delay signed, the project will not run or be debuggable.

2. Compile the project as a Release build.

3. Copy the HeaderBlock.dll binary from the Release directory into the GAC directory. The GAC directory can be found at the following location:

   C:\Windows\Microsoft.NET\assembly\GAC_32

Configuration

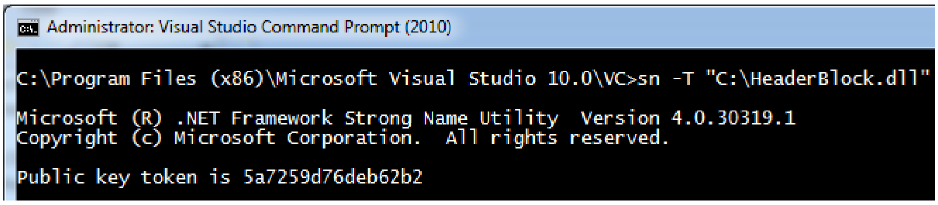
Once the module is installed into the GAC, it is simply a case of modifying either the web.config for a specific application or it can be performed on a machine basis. It is recommended that the module is used on a per machine basis.

When modifying the configuration files, the **PublicKeyToken** must be identified. If the Inno Setup installer is used then the **PublicKeyToken** value of **5a7259d76deb62b2** should be used to replace the **ABCDEF** place marker value used below.

If the module has been compiled by the user, then the Visual Studio **sn.exe** command line tool can used to extract the Public Key Token value. The Visual Studio command prompt must be opened to ensure that the **sn.exe** executable is on the command line path. The command line required to retrieve the Public Key Token value is shown below:

sn -T "PATH TO HeaderBlock.dll"

The screenshot below shows example output from the command:



```
C:\Program Files (x86)\Microsoft Visual Studio 10.0\VC>sn -T "C:\HeaderBlock.dll"

Microsoft (R) .NET Framework Strong Name Utility  Version 4.0.30319.1
Copyright (c) Microsoft Corporation.  All rights reserved.

Public key token is 5a7259d76deb62b2
```

If the module is being applied on an application basis then the applications configuration file (web.config) needs to be modified to include the following, replacing the PublicKeyToken value with the appropriate value:

```
<system.webServer>
  <modules runAllManagedModulesForAllRequests="true">
    <add name="HeaderBlock"
         type="ContextIS.HeaderBlock, HeaderBlock, Version=1.0.0.0, Culture=neutral,
PublicKeyToken= ABCDEF" />
  </modules>
</system.webServer>
```

To modify the configuration on a machine basis, the following file must be modified:

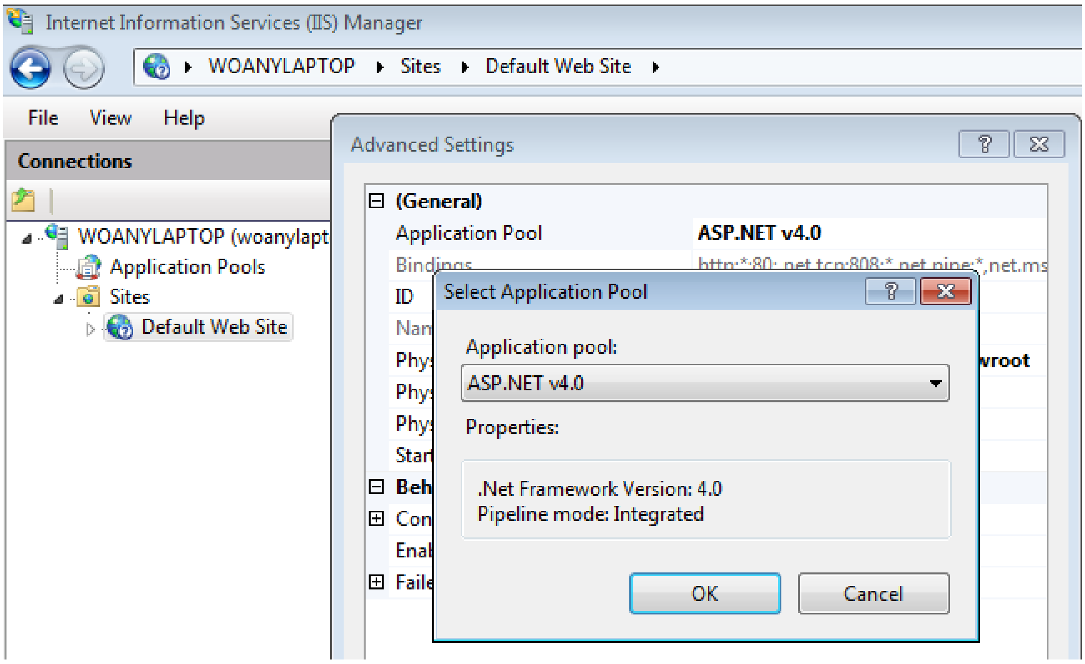
C:\Windows\System32\inetsrv\config\applicationHost.config

If the host operating system is x64 based, then the file must be modified using an x64 editor e.g. Notepad, since the file cannot be modified via an x86 editor due to file system redirects.

The configuration file (applicationHost.config) needs to be modified to include the following text within the <system.webServer><modules> section, replacing the PublicKeyToken value with the appropriate value:

```
<add name="HeaderBlock"
     type="ContextIS.HeaderBlock, HeaderBlock, Version=1.0.0.0, Culture=neutral,
PublicKeyToken=ABCDEF" />
```

When using the module the web server or application must be changed to use the Integrated .Net v4 Application Pool, depending on whether the module is being used on a machine or application basis. The following screen shot shows the Application Pool configuration window within the IIS Manager application:



Application Pool: ASP.NET v4.0

Bindings: http:*:80:,net.tcp:808:*,net.pipe:*,net.ms

Application pool: ASP.NET v4.0

Properties:

.Net Framework Version: 4.0
Pipeline mode: Integrated

Version History

v1.0.2

- Revised documentation re Ben Heinkel comments (Mark Woan)

v1.0.1

- Revised documentation re Ben Heinkel comments (Mark Woan)

v1.0.0

- Initial release (Mark Woan)

Context Information Security Ltd

London (HQ)
4th Floor
30 Marsh Wall
London E14 9TP
United Kingdom

Cheltenham
Corinth House
117 Bath Road
Cheltenham GL53 7LS
United Kingdom

Düsseldorf
Adersstr. 28, 1.OG
D-40215 Düsseldorf
Germany