

Microsoft® Virtual Labs

URL Authorization and Request Filtering in IIS 7

Microsoft®

Table of Contents

Exercise 1 URL Authorization	1
Exercise 2 Request Filtering.....	5

URL Authorization and Request Filtering in IIS 7

Objectives

Authorization was always a bit tricky in previous versions of IIS. Because IIS only worked with Windows® identities, you had to go to the file system and set Access Control Lists (ACL) on files and directories. This was tedious because the ACL UI is complex and authorization rules don't copy very well from machine to machine.

IIS 7 introduces URL Authorization. It allows you to put authorization rules on the actual URL instead of the underlying file system resource. On top of that, IIS 7 URL Authorization configuration is stored in web.config files, i.e. you can distribute your authorization rules with your application content. The following walkthrough introduces you to the IIS 7 URL Authorization feature in Windows Server® 2008 Beta 3 and Windows Vista™ Service Pack 1.

Estimated Time to Complete This Lab

45 Minutes

Computers used in this Lab



ContosoWeb1



The password for the Administrator account on all computers in this lab is:
pass@word1

Exercise 1

URL Authorization

Scenario

Let's simulate a scenario where you have a secure directory that only Alice, Bob, and the Administrators group can access. Within this directory we will have a file called bobsSecret.aspx that only Bob is supposed to access.

Tasks	Detailed Steps
<p>Complete the following tasks on:</p>  ContosoWeb1 <p>1. Launch the Microsoft® Windows Server® 2008 virtual machine and log on</p>	<p>a.  If the ContosoWeb1 virtual machine is not already running, start it using Virtual PC on the physical host computer.</p> <p>b. Press RIGHT ALT+DELETE to launch the Logon dialog box.</p> <p>c. Log on to the ContosoWeb1 Virtual PC with the following credentials: User name: Administrator Password: pass@word1</p> <p>d. Click OK.</p> <p><i>Note: You may enter full-screen mode by pressing RIGHT ALT+ENTER. You may exit full-screen mode at any point by pressing the key combination again.</i></p>
<p>2. Create sample users and set up a group</p>	<p><i>Note: For this scenario we need three users: Alice, Bob, and Fred. We also need a new group called BobAndFriends in which Alice and Bob are members.</i></p> <p>a. Click Start, right-click Command Prompt, and then click Run as administrator.</p> <p>b. Enter the following commands to populate the test users and groups that will be used in this exercise. Press Enter to execute the command before moving on to the next line. **QuickCommandsText**</p> <pre>net user Alice pass@word1 /add net user Bob pass@word1 /add net user Fred pass@word1 /add net localgroup BobAndFriends /add net localgroup BobAndFriends Alice /add net localgroup BobAndFriends Bob /add</pre> <p>c. Close the Command Prompt.</p>
<p>3. Create a new folder in wwwroot</p>	<p>a. Click Start, type explorer into the Start Search box, and then click Windows® Explorer.</p> <p>b. Navigate to C:\inetpub\wwwroot, right-click in the right pane, select New, Folder, and enter secure as the name.</p>
<p>4. Create default.aspx and enter source code</p>	<p>a. Click Start, type notepad into the Start Search box, and then click Notepad.</p> <p>b. Enter the following code into the text editor, which will comprise the default.aspx file for the secure folder: **QuickCommandsText**</p> <pre><%@Language="C#"%> <% string currentUser = Request.ServerVariables["LOGON_USER"]; if (currentUser == "") currentUser = "anonymous";</pre>

Tasks	Detailed Steps
	<p>Response.Write("Current User: " + currentUser + " "); %></p> <p>c. Press CTRL+S, click Browse Folders, navigate to C:\inetpub\wwwroot\secure, enter default.aspx as the file name, and then click Save.</p> <p>d. Close Notepad.</p>
<p>5. Create bobsSecret.aspx and enter source code</p>	<p>a. Click Start Notepad.</p> <p>b. Enter the following code into the text editor, which will comprise the bobsSecret.aspx file for the secure folder: <i>**QuickCommandsText**</i></p> <pre><% @Language="C#"%> <% string currentUser = Request.ServerVariables["LOGON_USER"]; if (currentUser == "") currentUser = "anonymous"; Response.Write("Current User: " + currentUser + " "); Response.Write("My secret: I used Apache before I discovered IIS7. "); %></pre> <p>c. Press CTRL+S, click Browse Folders, navigate to C:\inetpub\wwwroot\secure, enter bobsSecret.aspx as the file name, and then click Save.</p> <p>d. Close Notepad.</p>
<p>6. Confirm that the new files are functioning correctly</p>	<p>a. Click Start, then click All Programs, and then click Windows® Internet Explorer®.</p> <p>b. Navigate to http://localhost/secure and confirm that the Current User displays as anonymous.</p> <p>c. Navigate to http://localhost/secure/bobsSecret.aspx and confirm that the Current User still displays as anonymous, and that Bob's secret is also shown.</p> <p>d. Close Internet Explorer.</p>
<p>7. Enable authentication</p>	<p><i>Note: Authentication answers the question "who" wants to have access. Authorization answers "if" the authenticated "who" actually gets access. So before we investigate URL authorization we have to enable authentication because without knowing "who" wants to have access we can't answer the "if".</i></p> <p>a. To launch Internet Information Services Manager, click Start, and then click Internet Information Services (IIS) Manager.</p> <p>b. Expand the machine node CONTOSOWEB1, expand Sites, expand Default Web Site, and select the secure folder.</p> <p>c. In the Feature pane, double-click Authentication.</p> <p>d. Select Anonymous Authentication, and then in the Actions pane, click Disable.</p> <p>e. Select Basic Authentication, and then in the Actions pane, click Enable.</p>
<p>8. Verify that basic authentication is in place and functional</p>	<p>a. Launch Internet Explorer.</p> <p>b. Navigate to http://localhost/secure.</p> <p>c. In the Connect to localhost credentials window, authenticate as the user Alice</p>

Tasks	Detailed Steps
	<p>which you created earlier.</p> <p>User name: Alice</p> <p>Password: pass@word1</p> <p><i>Note: Notice that the Current User is now reflecting Alice as per basic authentication.</i></p> <p>d. Close Internet Explorer to clear cached credentials.</p> <p>e. Launch Internet Explorer again and navigate to http://localhost/secure/bobsSecret.aspx.</p> <p>f. In the Connect to localhost credentials window, authenticate as the user Fred which you created earlier.</p> <p>User name: Fred</p> <p>Password: pass@word1</p> <p><i>Note: You should see that Fred can access Bob's secret page using basic authentication.</i></p> <p>g. Close Internet Explorer to clear cached credentials.</p>
<p>9. Secure the pages to limit access by group</p>	<p>a. Return to the Internet Information Services (IIS) Manager window.</p> <p>b. Click the secure folder in the left pane to return to the secure Home view.</p> <p>c. Double-click Authorization Rules in the Feature pane.</p> <p>d. Select the Allow All Users rule.</p> <p>e. In the Actions pane, click Remove.</p> <p>f. In the Confirm Remove window, click Yes to disable the access to all users.</p> <p>g. In the Actions pane, click Add Allow Rule.</p> <p>h. In the Add Allow Authorization Rule window, select the Specified roles or user groups radio button, and enter BobAndFriends in the text box.</p> <p>i. Click OK. This enables access to the group you created earlier, which contains Alice and Bob, but not Fred.</p>
<p>10. Test the new security settings</p>	<p>a. Launch Internet Explorer.</p> <p>b. Navigate to http://localhost/secure.</p> <p>c. In the Connect to localhost credentials window, try to authenticate as Fred.</p> <p>User name: Fred</p> <p>Password: pass@word1</p> <p><i>Note: Since Fred is not in the BobAndFriends group, access is denied and the credentials dialog reappears.</i></p> <p>d. In the Connect to localhost credentials window, instead authenticate as Alice.</p> <p>User name: Alice</p> <p>Password: pass@word1</p> <p><i>Note: Alice can access the page since she is in the BobAndFriends group. Bob will also be able to access the site since he is in the same group. You can test this by closing Internet Explorer to clear Alice's credentials and then logging in as Bob.</i></p> <p>e. Close Internet Explorer to clear cached credentials.</p>

Tasks	Detailed Steps
<p>11. Configure URL authorization for a single Web page</p>	<p><i>Note: We still have the problem that Alice can access Bob's secret page (bobsSecret.aspx). We need to adjust URL authorization so that only Bob can access that page.</i></p> <ol style="list-style-type: none"> a. Return to the Internet Information Services (IIS) Manager window. b. In the left pane, click the secure folder to return to the Home view. c. In the Feature pane, click Content View near the bottom of the screen. You will see a list of files in the secure folder. d. Right-click bobsSecret.aspx, and then in the Actions pane, and then click Switch to Features View. <p><i>Note: You are now at the Home view for bobsSecret.aspx. Any changes made here will apply only to that page.</i></p> <ol style="list-style-type: none"> e. In the Feature pane, double-click Authorization Rules. You will see the Allow rule created earlier that bobsSecret.aspx has inherited from the security folder. f. Select the Allow BobAndFriends rule and then, in the Actions pane, click Remove. g. In the Confirm Remove window, click Yes to remove the existing rule. h. In the Actions pane, click Add Allow Rule. i. In the Add Allow Authorization Rule window, select the Specified users radio button, and enter Bob in the text box. j. Click OK. This restricts access to only Bob.
<p>12. Confirm that only Bob can access bobsSecret.aspx</p>	<ol style="list-style-type: none"> a. Launch Internet Explorer. b. Navigate to http://localhost/secure/bobsSecret.aspx. c. In the Connect to localhost credentials window, try to authenticate as Alice. <p style="margin-left: 40px;">User name: Alice</p> <p style="margin-left: 40px;">Password: pass@word1</p> <p><i>Note: Since the BobAndFriends group rule was removed from bobsSecret.aspx, access is denied and the credentials dialog reappears.</i></p> <ol style="list-style-type: none"> d. In the Connect to localhost credentials window, instead authenticate as Bob. <p style="margin-left: 40px;">User name: Bob</p> <p style="margin-left: 40px;">Password: pass@word1</p> <p><i>Note: Now only Bob can now access the site, and his secret is safe from Alice and Fred.</i></p> <ol style="list-style-type: none"> e. Close Internet Explorer.

Exercise 2

Request Filtering

Scenario

URLScan was a security tool that was provided as an add-on to earlier versions of IIS so administrators could enforce tighter security policies on their Web servers. Within IIS 7 we have incorporated all the core features of URLScan into a module called Request Filtering and added an additional feature called Hidden Segments. This exercise recaps each of the features that Request Filtering provides and gives a real world example of how you can apply it in your environment.

Tasks	Detailed Steps
<p>1. Examine hidden segments in applicationhost.config</p>	<p>a. Launch Internet Explorer.</p> <p>b. Navigate to http://localhost/secure/web.config.</p> <p><i>Note: Notice that you cannot access the file, and you receive a 404.8 error. The request filtering module is configured to deny a path in the URL that contains a <hiddenSegments> section in the configuration file. See note in Step 7 for additional information.</i></p> <p>c. Close Internet Explorer.</p> <p>d. Switch to Windows Explorer.</p> <p>e. Navigate to C:\Windows\System32\inetrv\config.</p> <p>f. Right-click applicationHost.config and select Edit.</p> <p><i>Note: IIS 7 allows you to define servable segments within configuration files and returns an error code logged as 404.8 when IIS 7 rejects a request for the defined segment feature.</i></p> <p><i>Using the Go To... (Ctrl+G) option in Notepad, search between lines 360 and 417 in the applicationHost.config file to view the <requestFiltering> section and identify the various sections, including where web.config is not allowed. Also view the hidden namespaces section and observe that specific locations are not allowed in a URL such as BIN and _APP_DATA>.</i></p> <p>g. Close Notepad.</p>
<p>2. Hidden segment implementation using web.config</p>	<p>a. Launch a Command Prompt.</p> <p>b. Type the following commands to create a new folder and populate it with one of the aspx files created in the previous exercise: **QuickCommandsText**</p> <pre>cd \inetpub\wwwroot md secrets cd secure copy bobsSecret.aspx ..secrets exit</pre> <p>c. Launch Internet Explorer.</p> <p>d. Navigate to http://localhost/secrets/bobsSecret.aspx. Notice that the file is now accessible anonymously.</p> <p>e. Close Internet Explorer.</p> <p>f. Click Start, and then click Notepad.</p> <p>g. Enter the following code into the text editor, which will comprise the web.config</p>

Tasks	Detailed Steps																		
	<p>file for the secrets folder: **QuickCommandsText**</p> <pre data-bbox="581 268 1052 646"><?xml version="1.0" encoding="UTF-8"?> <configuration> <system.webServer> <security> <requestFiltering> <hiddenSegments> <add segment="secrets"/> </hiddenSegments> </requestFiltering> </security> </system.webServer> </configuration></pre> <p>h. Press CTRL+S, click Browse Folders, navigate to C:\inetpub\wwwroot\secrets, enter web.config as the file name, and then click Save.</p> <p>i. Close Notepad.</p> <p><i>Note: In the delegated configuration architecture system used in IIS 7, the rules for securing the site are stored with the Web content. This enables deployment to different sites and servers without having to repeat any server configuration.</i></p> <p>j. Launch Internet Explorer.</p> <p>k. Navigate to http://localhost/secrets/bobsSecret.aspx again to test the hidden segment code added to the web.config file.</p> <p><i>Note: Note that you receive the HTTP Error 404.8 which describes the hiddenSegment path deny that has been enforced.</i></p> <p><i>Note: If the new web.config seems to have no effect, with bobsSecret.aspx displaying normally, restart IIS using the IIS Manager and try again.</i></p> <p><i>The following table shows each request filtering error message and the corresponding codes for easy reference.</i></p> <table border="1" data-bbox="506 1247 1334 1843"> <thead> <tr> <th>Error</th> <th>Status Codes</th> </tr> </thead> <tbody> <tr> <td>Site not found</td> <td>404.1</td> </tr> <tr> <td>Denied by policy</td> <td>404.2</td> </tr> <tr> <td>Denied by mime map</td> <td>404.3</td> </tr> <tr> <td>No handler</td> <td>404.4</td> </tr> <tr> <td>Request Filtering: URL Sequence denied</td> <td>404.5</td> </tr> <tr> <td>Request Filtering: Verb denied</td> <td>404.6</td> </tr> <tr> <td>Request Filtering: File extension denied</td> <td>404.7</td> </tr> <tr> <td>Request Filtering: Denied by hidden segment</td> <td>404.8</td> </tr> </tbody> </table>	Error	Status Codes	Site not found	404.1	Denied by policy	404.2	Denied by mime map	404.3	No handler	404.4	Request Filtering: URL Sequence denied	404.5	Request Filtering: Verb denied	404.6	Request Filtering: File extension denied	404.7	Request Filtering: Denied by hidden segment	404.8
Error	Status Codes																		
Site not found	404.1																		
Denied by policy	404.2																		
Denied by mime map	404.3																		
No handler	404.4																		
Request Filtering: URL Sequence denied	404.5																		
Request Filtering: Verb denied	404.6																		
Request Filtering: File extension denied	404.7																		
Request Filtering: Denied by hidden segment	404.8																		

Tasks	Detailed Steps														
	<table border="1"> <tr> <td data-bbox="509 197 1166 256">Denied since hidden file attribute has been set</td> <td data-bbox="1166 197 1334 256">404.9</td> </tr> <tr> <td data-bbox="509 256 1166 352">Request Filtering: Denied because request header is too long</td> <td data-bbox="1166 256 1334 352">404.10</td> </tr> <tr> <td data-bbox="509 352 1166 411">Request Filtering: Denied because URL doubled escaping</td> <td data-bbox="1166 352 1334 411">404.11</td> </tr> <tr> <td data-bbox="509 411 1166 470">Request Filtering: Denied because of high bit characters</td> <td data-bbox="1166 411 1334 470">404.12</td> </tr> <tr> <td data-bbox="509 470 1166 529">Request Filtering: Denied because content length too large</td> <td data-bbox="1166 470 1334 529">404.13</td> </tr> <tr> <td data-bbox="509 529 1166 588">Request Filtering: Denied because URL too long</td> <td data-bbox="1166 529 1334 588">404.14</td> </tr> <tr> <td data-bbox="509 588 1166 680">Request Filtering: Denied because query string too long</td> <td data-bbox="1166 588 1334 680">404.15</td> </tr> </table>	Denied since hidden file attribute has been set	404.9	Request Filtering: Denied because request header is too long	404.10	Request Filtering: Denied because URL doubled escaping	404.11	Request Filtering: Denied because of high bit characters	404.12	Request Filtering: Denied because content length too large	404.13	Request Filtering: Denied because URL too long	404.14	Request Filtering: Denied because query string too long	404.15
Denied since hidden file attribute has been set	404.9														
Request Filtering: Denied because request header is too long	404.10														
Request Filtering: Denied because URL doubled escaping	404.11														
Request Filtering: Denied because of high bit characters	404.12														
Request Filtering: Denied because content length too large	404.13														
Request Filtering: Denied because URL too long	404.14														
Request Filtering: Denied because query string too long	404.15														
<p>3. Use URL length to restrict incoming requests</p>	<p><i>Note: You can use request filtering to restrict incoming requests based on URL length.</i></p> <ol style="list-style-type: none"> a. Switch to Windows Explorer. b. Navigate to C:\inetput\wwwroot\secrets. c. Double-click web.config. d. In Notepad, make a new line under the <code></hiddenSegments></code> tag (line 8) and enter the following code: **QuickCommandsText** <pre><requestLimits maxAllowedContentLength="3000000" maxUrl="260" maxQueryString="25" /></pre> <p><i>Note: The maxUrl attribute restricts the URL length to secure the server. Let's reduce this amount to observe it in action.</i></p> <ol style="list-style-type: none"> e. Still in Notepad, adjust the maxUrl value from 260 to 1. f. Press CTRL+S to save the changes. g. Switch to Internet Explorer and refresh the http://localhost/secrets/bobsSecret.aspx page. <p><i>Note: You will receive an HTTP Error 404.14 which describes the URL length request filtering deny rule that you just configured. Filtering by URL length can be a very strong defense against potentially unknown future attacks.</i></p> <ol style="list-style-type: none"> h. Switch to Notepad and adjust the maxUrl value to 100. i. Next, delete the hidden segments section: <pre><hiddenSegments> <add segment="secrets"/> </hiddenSegments></pre> j. Press CTRL+S to save the changes. k. Return to Internet Explorer and refresh the http://localhost/secrets/bobsSecret.aspx page again. <p><i>Note: You will observe that the page is now functional again after removing the maxUrl and hiddenSegment restrictions.</i></p>														

Tasks	Detailed Steps
<p>4. Use DenyURLSequence to prevent users from accessing a system folder</p>	<p>a. Switch to Notepad and add the following code below the requestLimits section in the requestFiltering block (line 10). <i>**QuickCommandsText**</i></p> <pre><denyUrlSequences> <add sequence="vti_bin"/> </denyUrlSequences></pre> <p><i>Note: This will deny any URL containing the character sequence vti_bin, which is a folder present on servers that have FrontPage Extensions installed.</i></p> <p>b. Press CTRL+S to save the changes.</p> <p>c. Switch to Internet Explorer and attempt to navigate to http://localhost/secrets/vti_bin.</p> <p><i>Note: You receive HTTP Error 404.5 which filters the request based on the specified URL sequence.</i></p>