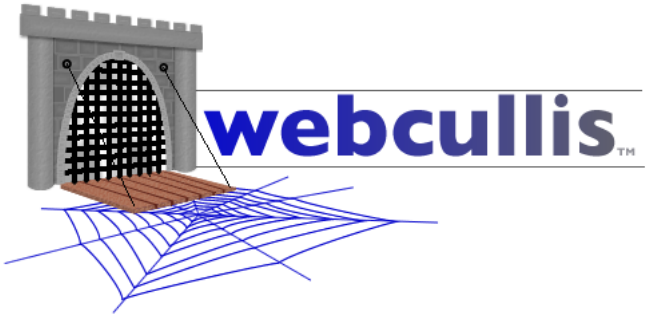
Webcullis™ Configuration Manual
April 2007

Cygnacom Solutions, Inc.



Contents

1 Introduction

Webcullis™[1] is a security plug-in for multiple https servers. It is designed to strengthen the web server's ability to limit access to files based on certificate policy or name constraints when the server implements X.509 PKI-based authorization schemes.

1.1 Overview

The heart of the Webcullis plug-in is its configuration file, which must be written according to the authorization policies of the web server in which it is being installed. The Webcullis installation includes a sample configuration file which may be useful in this writing. Webcullis access constraints are implemented on a per-directory basis and managed in this configuration file as described in Sections 4 and 5. We recommend that administrators read carefully the options for configuration before writing their configuration file, as improper access policies can lead to the compromise of otherwise secure systems.

In this document, we will first outline the procedure for installation of the Webcullis plug-in on a machine running IIS (Section 2). Next we will consider the options available for the configuration of Webcullis, including general options as well as those for per-directory access control (Section 4). Finally, we will consider excerpts of example configuration files to better illustrate the capabilities of Webcullis (Section 5), and conclude with information on resources for administrators using Webcullis (Section 6). Installation instructions for other servers will be covered in supplemental documentation included with the appropriate Webcullis distribution for those servers.

2 Installation on IIS Servers

To perform the installation with the Webcullis installer, complete the following steps, which are outlined in more detail in the rest of this section. This instructions assume that IIS has been installed and is configured to accept ISAPI filter plug-ins.

1. Run the Webcullis installer.
2. Write a configuration file.
3. Import trust anchors into the trustroot directory.
4. Configure ISAPI Filter for IIS.
5. Restart the web server.

[1] Webcullis™ is a trademark of Cygnacom Solutions, all rights reserved. All other trademarks and registered trademarks are the property of their respective owners. Unless stated to the contrary, no association with any other company or product is intended or should be inferred.

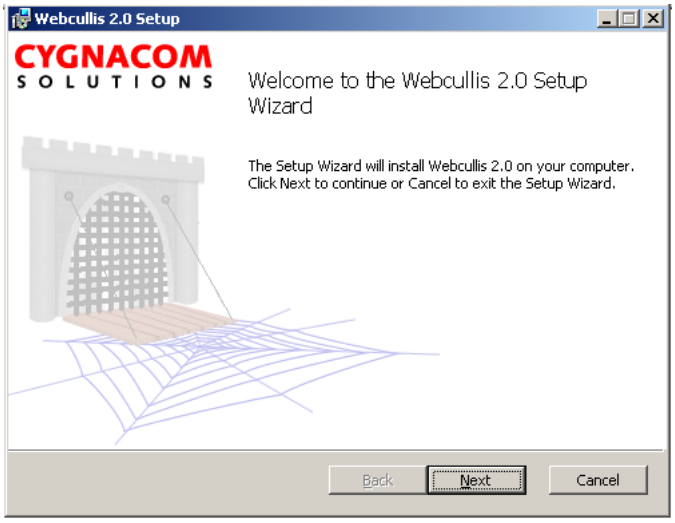
2.1 The Installer

Webcullis for IIS includes an installer which can be used to ease configuration of your server. The installer will perform the following actions:

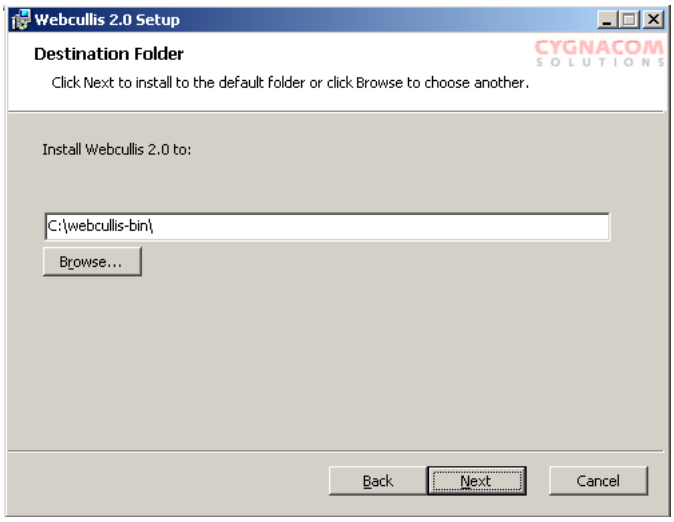
1. Copy the Webcullis files to your filesystem
2. Add the registry entry needed for Webcullis to find its .ini file
3. Create a directory for your trusted root certificates
4. Write a bare-bones initial configuration file
5. (Optional) Install the Webcullis graphical configuration utility
6. (Optional) Enable Webcullis globally for your IIS server

Note that the installer is not strictly necessary for the configuration or operation of Webcullis. Any of these steps can be performed manually or by a script if that makes more sense for deployment in a particular environment. Should that be preferable, download the .zip distribution of Webcullis instead of the .msi. Supplemental instructions for manual installation are included inside the .zip file.
Important Note for manual installations: Windows versions of Webcullis require version 8 or later of the Microsoft C++ runtime libraries. If you're performing a manual install, make sure those are present on your server. If they're missing, or if you have problems enabling Webcullis, they can be downloaded from http://www.orionsec.com/files/CRTSetup.msi

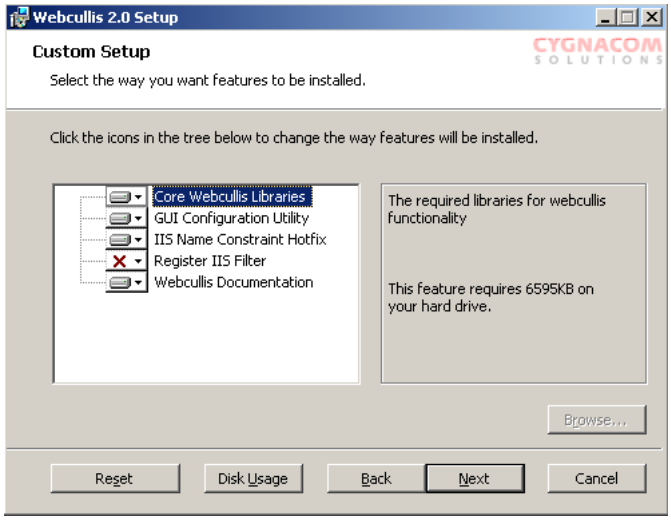
- Launch the Webcullis installer



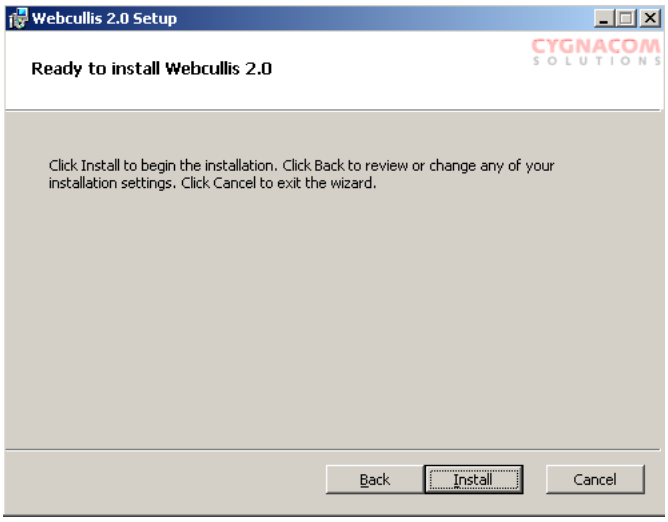
- Select an installation location If you change this location, be sure to change any example configurations you use accordingly.



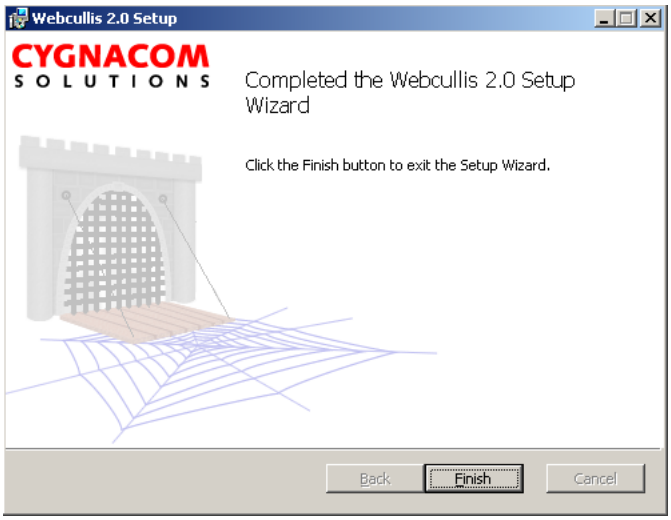
- Select installation options **Warning:** If the ”Register IIS Filter” option is selected, Webcullis will be enabled immediately for all virtual servers on the system. This will most likely break a production server until Webcullis is completely configured, and should only be selected in instances where downtime is permissible. Be sure to configure Webcullis and restart IIS immediately after the installation has completed if you select this option to minimize potential downtime.



- Confirm Installation



- Finish and continue



Now that the setup has completed, a few configuration tasks remain before Webcullis will be useful. First, edit the default configuration. Either use any standard text editor or use the new (and still somewhat experimental) GUI utility included with Webcullis. Then you'll need to copy any required trust anchors into the configured trust root directory, configure any desired restrictions or other settings, and prepare your web server. If you did not choose to enable Webcullis within IIS when prompted by the installer, you'll need to do so from the server's administrative interface.

2.2 The Configuration File

The sample configuration file provided with Webcullis can serve as a starting point in writing a custom version. See also Section 4 for more information on the options available for use in this file, and Section 5 for sample configuration files.

2.3 Trust Anchor Management

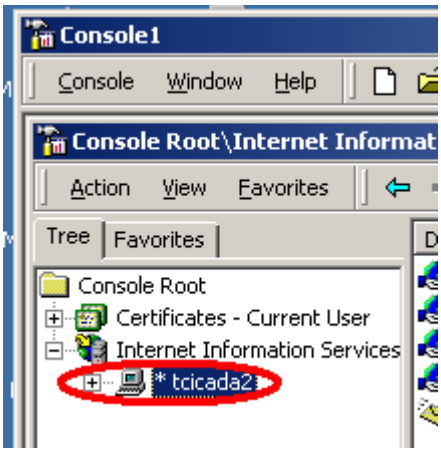
Webcullis maintains its own store of trusted root CA certificates distinct from those in the Windows CAPI store. These trust anchors are added or removed by copying them to or deleting them from the trustroot directory, which is identified in the configuration file. The contents of the directory are read every time Webcullis is restarted; it is recommended that all instances of IIS be restarted if modifications to this directory are made.

Because improper configuration of Webcullis or incorrect file permissions on the trustroot directory could open the web server to unauthorized access, we recommend that the administrator in charge of the web server on which Webcullis is to be installed first consult resources such as the National Security Agency's "Guide to the Secure Configuration of Microsoft Internet and Information Services" (Section 6). We note also that the Webcullis plug-in only needs read access to the files in the trustroots directory, and recommend that administrators apply permissions to those files accordingly.

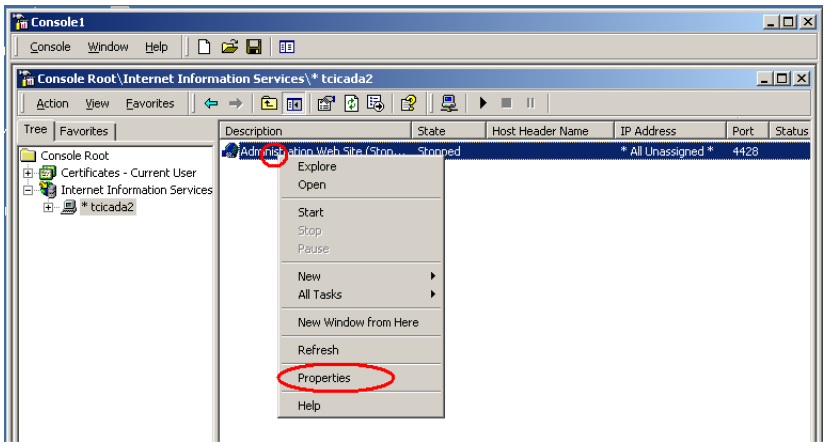
2.4 The ISAPI Filter

The Webcullis installer will include an ISAPI filter DLL, which is made available to IIS using the IIS Administration MMC snap-in. Follow these steps to configure IIS 5.0 to use the filter. These same steps work for IIS 6.0, provided you place the server into IIS 5-compatible worker process isolation mode instead of its default application pool mode. If the compatibility mode is not acceptable for your installation, see the supplemental information for IIS 6 for details on how to configure permissions so that the default mode works.

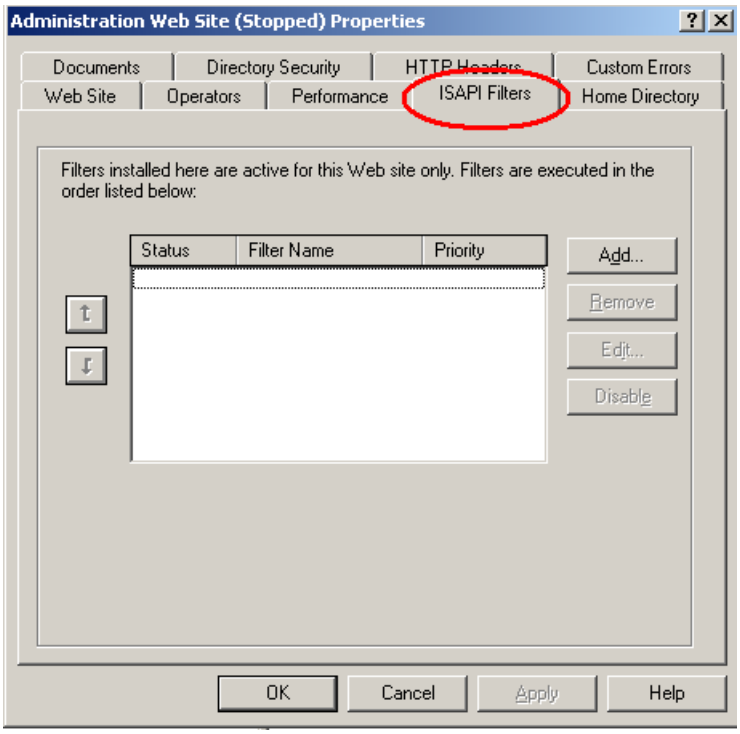
- From the start menu, choose Programs > Administrative tools > IIS.msc
- Within the tree window to the left, expand the Internet Information Services node to view the server machine name. Select this machine name.



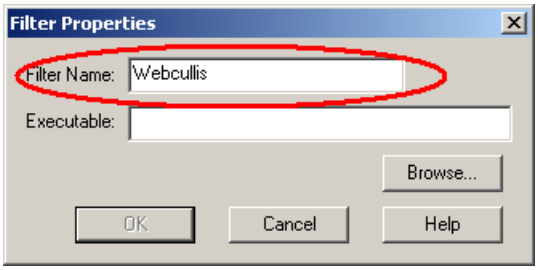
- Within the main window, identify the web server on which you would like to run Webcullis. Right click on the server name and choose properties.



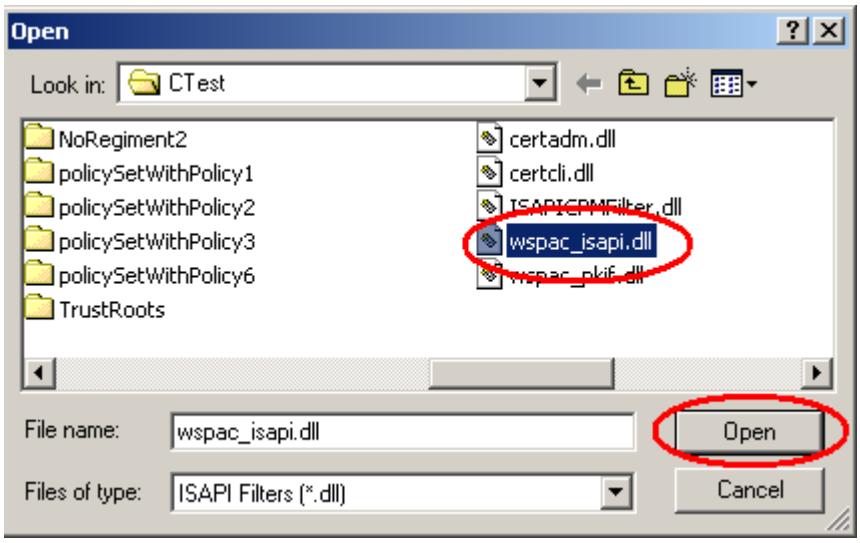
- Choose the ISAPI Filters tab.



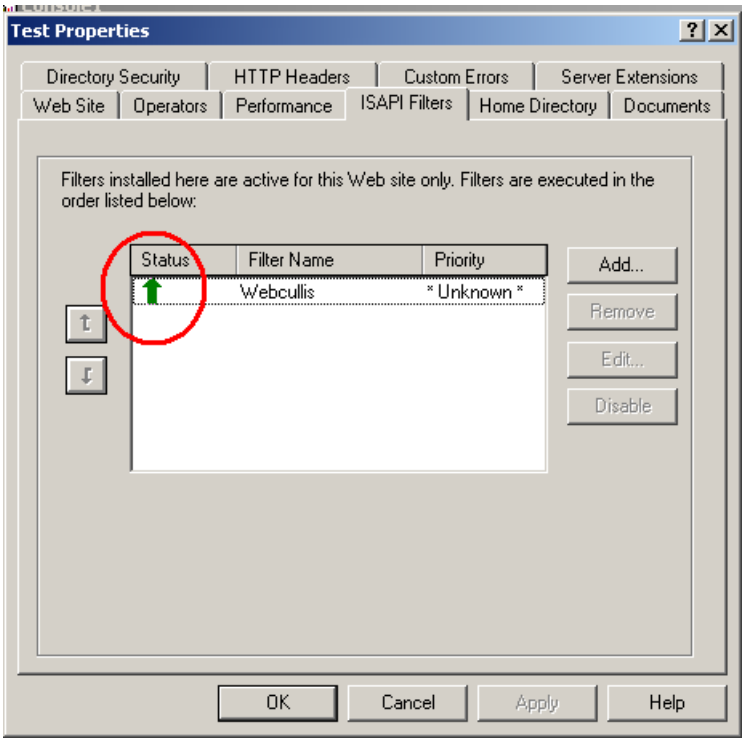
- Press the Add button. In the window that appears, enter a name to identify the plug-in, such as “Webcullis.”



- Press the browse button and navigate to the location of the Webcullis installation. Choose the file `wspac_isapi.dll` and press Open.



- Press OK in the Filter Properties window.
- Press OK in the properties window of the web server.
- To start using the Webcullis plug-in, restart the web server. This is accomplished by right-clicking on the server name in the main body of the window and choosing Stop, then Start.
- Verify that Webcullis is running by again right-clicking on the web server, choosing properties, and clicking on the ISAPI Filters tab. To the left of the Webcullis filter name there should be a green arrow indicating that it is running.



2.5 De-installation

To de-install Webcullis, you should first remove it from IIS. This is accomplished by at first following the instructions of Section 2.4 on how to configure it as an ISAPI filter. However, where the directions instruct you to add the new filter, you should instead select the Webcullis filter and remove it. For your changes to take effect, you should restart all web servers that were using Webcullis for access control.

Once Webcullis is no longer running, you can safely remove the software itself from your computer. If the installer was used, simply use the Add/Remove Software control panel to remove Webcullis. Otherwise, remove the files and the registry entries you installed manually.

3 Installation on Linux

To perform the installation using Webcullis Linux distribution tarball, complete the following steps. These instructions assume that you have already configured client SSL authentication for your apache install and that it works properly.

1. Stop the apache service.
2. Create the directory `/usr/local/pkif`

3. Copy the lib directory into `/usr/local/pkif`
4. Edit `/etc/ld.so.conf` to contain a line that reads `/usr/local/pkif/lib`
5. Execute the `ldconfig` command.
6. Copy the files from modules to `/usr/lib/httpd/modules` NOTE: This will overwrite your existing `mod_ssl.so`. Back it up first when needed.
7. Copy the `conf/webcullis` directory to `/etc/webcullis`
8. Modify `/etc/webcullis/webcullis.cf` as appropriate for your site.
9. Create the directory `/etc/webcullis/trustroots`
10. Copy DER-encoded certificates for your trusted root CAs into `/etc/webcullis/trustroots`
11. Edit `/etc/httpd/conf.d/ssl.conf` to activate webcullis. Examples of the commands required for this are in conf/ssl.conf.sample. Be sure to change your SSLVerifyClient setting to ”webcullis” to prevent mod_ssl bugs from interfering with path validation.
12. (Optional) If you are using an NSS database, be sure the NSSDatabase is properly configured in webcullis.cf and that it has been populated with trust anchors, otherwise webcullis will be unable to add intermediates and CRLs to the store. Required tools for creating and populating this database are included in bin.

    Certificates should be imported with ”C,,” trust flags.

    The following command will create a new nss database.

    cd into the bin folder and execute

    ```
    ./certutil -N -d /etc/webcullis/nssdb
    ```

    The following command will add a Trust Anchor to nss database

    cd into the bin folder and execute (for example)

    ```
    ./certutil -A -d /etc/webcullis/nssdb -n JITCTA -t "C,," -i JITCTA.crt
    ```

    After creating the database and addding the trust anchor add the NSSDatabase property to Webcullis config file.

    ```
    NSSDatabase = /etc/webcullis/nssdb
    ```
13. Start the apache service.

4 Configuration Options

The options used in the Webcullis configuration file can be broken into two broad categories: those that affect the Webcullis program as a whole, and those that affect the way in which authorization is decided for a given directory tree (for example, the set of directories and files rooted at `/reports/fy2000`).

Note that options of the second category– those that enforce authorization policies on a given directory– can also be specified as general options in the configuration file header. Restrictions on a directory will be executed in the following order: **Specific** (defined in an individual directory block of the configuration file), **General** (specified in the header block of the configuration file) and **Default** (no value is specified, and the default Webcullis value is used). Note also that specifying a general authorization option in the header has a different effect than specifying one for the root directory, which is discussed in Section 4.2.

Options are generally specified in the configuration file in the following format:

```
RequireRecentCRL = Yes
```

where the left hand side of the equation is the option key and the right had side is the value that the administrator would like implemented for this option. In this example, the administrator has decided to require a recent Certificate Revocation List while building a validation path for certificates presented for authorization to the given directory tree. Note that values of `yes`, `Yes`, and `YES` will be treated equally, whereas any other value to binary options will be treated as `No`.

See Section 5 for more information on configuration file formatting.

4.1 General Configuration Options

The following table outlines the Webcullis general configuration options:

Option Key	Valid Value(s)	Default Value
TrustRootPath	Directory	(none)
CacheEntries	0-2147483647	150
MaxCacheAge	0-2147483647 (seconds)	300
LogLevel	Log level	(none)
LogPath	Log file	(none)
TracePath	Log file for trace logs	(none)
ErrorDocument	HTML file	(none)
NSSDatabase	Directory	(none)
BlacklistedServer	Blacklisted Server	none

4.1.1 TrustRootPath

This is the location of the Webcullis trustroot store (see Section 2.3 for more information). The value assigned to this option should be a well-formatted absolute path pointing a directory containing only properly formatted DER-encoded certificate files. **This is a required field.** Without a value for this field in the configuration file, Webcullis will not be able to run. Note that the use of backslashes (\) is necessary to specify a path in the Windows file system.

4.1.2 CacheEntries

This is the number of certificate chain validation results that the server will cache. While the identifier for a given certificate is in the cache and younger than the value specified by `MaxCacheAge`, the server will not re-validate that cert chain when presented with it during an authorization request.

4.1.3 MaxCacheAge

The maximum number of seconds for which a certificate chain identifier will be cached.

4.1.4 LogLevel

This is how the verbosity of the log messages are configured. A value of 0 specifies no logging; 5 is the most verbose. It is strongly recommended that logging be set at least to level 1, and this setting should be sufficient in the general case. The following table outlines the information logged at each level:

Level	Verbosity
0	No messages will be logged after plug-in startup. (Not Recommended)
1	Only error messages will be logged.
2	Error and warning messages.
3	Error, warning and informational messages.
4	All of the above plus debug messages.
5	All of the above plus path building and validation traces.

4.1.5 LogPath

This is the location of the principal Webcullis log file in the Windows file system, specified using backslashes (\). All log messages generated below level 5 (trace level) will be logged to this file. **This is a required field**. Failing to set it in the configuration file will prevent Webcullis from starting.

It is recommended that system administrators monitor the size of this file during the initial period of operation and consider placing it under a regular automated rotation schedule. Its rate of growth will depend greatly on the amount of traffic to the web server and the configured verbosity of the Webcullis log messages.

4.1.6 TracePath

This is the file to which trace log messages will be written if the LogLevel option is set to 5. It is highly recommended that a system administrator delete or rotate this file off disk if trace logging is enabled, as it could grow very large in a short amount of time depending on the amount of traffic to the web server.

If no trace file is specified, no trace logging will be performed. If the log level is set to 5 but no trace file is specified, a message stating this fact will be written to the general log file.

4.1.7 ErrorDocument

This is the HTML file to which a browser requesting authorization will be directed if the validation fails. If no page is specified, the browser will be directed to an empty page. The sample configuration file points to a generic error document provided with the Webcullis installation.

4.1.8 NSSDatabase

This is the location of the NSS (Network Security Services) database. The value assigned to this option should be a well-formatted absolute path pointing to NSS database directory.

4.1.9 BlacklistedServer

The configuration file may list zero or more blacklisted servers. Each blacklisted server entry in the configuration file identifies an LDAP-accessible directory server that is known to be problematic, i.e., performs poorly, is no longer in service, etc. Such servers are often included in long-lived artifacts, such as public key certificates. Webcullis will not attempt to retrieve PKI artifacts from servers that appear on the blacklist. Each blacklisted server is listed in a separate line in the configuration file, as shown in the following snip:

```
BlacklistedServer = server1.example.com
BlacklistedServer = server2.example.com
```

4.2 Directory-Level Configuration Options

Directory-specific configurations are made in blocks starting with an identifier of the directory. So, a block containing configurations for the directory `/reports/fy00` would start with the following line:

```
[/reports/fy00]
```

Note: The root directory of the IIS web file system is considered as the root for the directory access control options. Forward slashes (/) should be used to identify all directories relative to the web server root for directory-level options. It is not recommended that files be referenced from outside this subtree in configuring directory-level access control policies. Do not attempt to construct absolute paths in the Windows file system.[2]

Note also that restrictions are inherited throughout the subtree of the directory to which they are applied. If `/reports` is restricted with the `RequireFreshCRL` option, then access to the `/reports/fy00` subdirectory would also be restricted. Note

[2] In rare cases where forward slashes are used as application-specific delimiters, absolute file names delimited by backslashes can be considered. However, this form is discouraged for the majority of Webcullis configurations.

also that once a restriction is tightened on a directory, it can not be loosened on a subtree of that directory; in the above example, creating a block for `/reports/fy00` and excluding the option `RequireFreshCRL` would have no effect because the `RequireFreshCRL` option had already been applied to `/reports`. This means that any restrictions applied to the root directory (`/`) will be applied to all files on the server, *even if looser restrictions are defined for subdirectories*. In Section 4 it is also noted that you can specify general configuration settings in the configuration file header that can be overridden at the individual directory level.

Option Key	Valid Value(s)	Default Value
RequireRecentCRL	Yes/No	No
CRLFreshness	0-2.1 billion (seconds)	0
RequireFreshCRL	Yes/No	No
AllowedPolicy	OID	(none)
RequireAllPolicies	Yes/No	No
InitialExplicitPolicy	Yes/No	No
InitialInhibitAnyPolicy	Yes/No	No
PolicyMapInhibit	Yes/No	No
ExtendedKeyUsage	OID	(none)
RequireMatchAllEKU	Yes/No	No
PermittedSubtree	Distinguished Name (DN)	(none)
ExcludedSubtree	Distinguished Name (DN)	(none)
MinKeySize	1-32767	1024
LDAP	server;port;namespaces	none
OCSP	url;multi;reserved;reserved;namespaces	none

4.2.1 RequireRecentCRL

This option allows a system administrator to require that a cached CRL be no older than a certain number of seconds in order for it to be used during validation. This time limit is specified in the `CRLFreshness` value. Note that this option is not enabled by default.

4.2.2 CRLFreshness

This option is used to specify the maximum age in seconds for a CRL used during validation. If a CRL is older than this limit, a fresh copy will be fetched before validation is performed. If this option is omitted or provided with a value of 0, it will be implemented as a 30 day limit by default.

4.2.3 RequireFreshCRL

If this option is used, a cached CRL is used during validation only if the current date is before the CRL's nextUpdate field. In this manner, a cached CRL is only updated if its issuer should have published an update by the time of the validation. Note that this option is not enabled by default.

4.2.4 AllowedPolicy

This option allows an administrator to specify a certificate policy that is acceptable for accessing the given directory. It is possible to list multiple allowed policies; an entry is required for each one:

```
AllowedPolicy=2.16.840.1.101.3.2.1.48.2
AllowedPolicy=2.16.840.1.101.3.2.1.48.6
...
```

If the `AllowedPolicy` option is used, a certificate must contain at least one of the allowed policies in order to be acceptable for access. **Note:** In order for the policies specified under this option to be enforced, the `InitialExplicitPolicy` option must also be set to `Yes`. In the vast majority of cases it will be desirable to also set the `InitialInhibitAnyPolicy` option to `Yes`.

4.2.5 RequireAllPolicies

This option is only used in a directory configuration block containing more than a single `AllowedPolicy` statement. It requires that an acceptable certificate contain *all* listed policies, instead of the default of ≥ 1.

4.2.6 InitialExplicitPolicy

This option indicates that the end entity certificate in the chain being validated must contain an explicit certificate policy. This option must be set to `Yes` to perform access control based on certificate policies.

4.2.7 InitialInhibitAnyPolicy

This allows a system administrator to ignore the assertion of the special "any policy" in the end entity certificate being validated. In this manner, if access is restricted based on certificate policies, it can only be granted if the certificate contains one (or all) of the exact policies put forth in the configuration file.

4.2.8 PolicyMapInhibit

This option allows a system administrator to prevent the validation of a certificate chain using a mapped policy. In this manner, if there are certificate policy restrictions using `AllowedPolicy`, they can only be satisfied if the certificate conforms to them natively.

4.2.9 ExtendedKeyUsage

This option is similar to the `AllowedPolicy` option, but instead of specifying certificate policy OIDs, it allows an administrator to limit access to a directory based on key usage extensions. As with `AllowedPolicy`, if the option is present at all in a

directory configuration block, a certificate must contain at least one of the key usage extensions listed to be acceptable.

4.2.10 RequireMatchAllEKU

This option is parallel to `RequireAllPolicies`. It allows an administrator to require that an acceptable certificate contain all specified key usage policies, instead of the default ≥ 1.

4.2.11 PermittedSubtree

This option, combined with `ExcludedSubtree`, allows an administrator to exert name restrictions during authorization. More than one permitted subtree can be specified with this option, in the manner used for the `AllowedPolicy` option.

If this option is used, only certificates with DNs falling under one of the listed subtree(s)q will be authorized for access. This option can be used in tandem with the `ExcludedSubtree` option, although this would be redundant unless the subtree listed in one was an ancestor to the subtree listed in the other. See Section 5.2 for an example.

Note that for both this and the `ExcludedSubtree` option, **the DN is specified in local order**, i.e., the most local qualifier is first:

`PermittedSubtree=cn=Jose`Vidro, o=State Polytechnic Institute, cu=us

4.2.12 ExcludedSubtree

This and the previous option allow a system administrator to use name constraints to control access to the web server file system. If this option is used, all certificates other than those listed under it will be allowed to access the given directory. As with `PermittedSubtree`, it is possible to list more than one excluded subtree.

4.2.13 MinKeySize

This option allows an administrator to limit access to certificates of keys with a certain size or greater. It should be noted that if this option is used, *the constraint will be applied to all certificates in a chain*, not just the end entity certificate. Thus, if a minimum key size of 1024 is imposed and the end entity certificate satisfies this requirement but another certificate in the chain does not, validation will fail.

4.2.14 LDAP

Zero or more LDAP entries may appear in the configuration file. Each identifies an LDAP-accessible directory server that will be consulted for certificates and CRLs as necessary. Each entry is formated as follows:

```
LDAP=<server>;<port>;<namespaces>
```

server is the DNS name or IP address of the directory server. This value must be present.

port is the port on which the server listens for LDAP connections. This value must be present.

namespaces is a semi-colon delimited list of base64-encoded GeneralNames. The Webcullis Configuration Utility can be used to add LDAP entries that include namespaces to the configuration file. This value is optional. If absent, the LDAP entry must end with a semi-colon, as shown in the example below. Multiple namespaces may appear separated by a semi-colon. The directory server will only be consulted for for certificates or CRLs with a subject name or issuer name, respectively, within one of the specified namespaces. If no namespace is specified, the responder will be consulted for each certificate validated by Webcullis.

```
LDAP = someserver1;389;
LDAP = someserver2;389;MCAxCzAJBgNVBAYTAlVTMREwDwYDVQQK...
```

4.2.15 OCSP

Zero or more OCSP entries may appear in the configuration file. Each identifies an OCSP responder that will be consulted for revocation status when validating certificates. Each entry is formated as follows:

```
OCSP=<server>;<multiCert>;<reserved>;<reserved>;<namespaces>
```

server is the DNS name or IP address of the directory server. This value must be present.

multiCert is a boolean value (true or false) that indicates whether multiple certificates should be included in OCSP requests sent to this server. false is the recommended value for maximum interoperability.

reserved is a placeholder for configuration options that may be added in the future. These values must be present.

namespaces is a semi-colon delimited list of base64-encoded GeneralNames. The Webcullis Configuration Utility can be used to add OCSP entries that include namespaces to the configuration file. This value is optional. If absent, the OCSP entry must end with a semi-colon, as shown in the example below. Multiple namespaces may appear separated by a semi-colon. The OCSP responder will only be consulted for revocation status of certificates containing a subject name within one of the specified namespaces. If no namespace is specified, the responder will be consulted for each certificate validated by Webcullis.

```
OCSP = someresponder1;false;reserved;reserved;
OCSP = someresponder2;false;reserved;reserved;MCAxCzAJBg...
```

5 Configuration File Examples

In this section we will consider a couple of simple example configuration file excerpts.

5.1 General Options: The Configuration File Header

Every Webcullis configuration file must be headed by a set of general configuration file options. Of these, `TrustRootPath` and `LogPath` are mandatory, and customizing `LogLevel` is strongly recommended. Other options need to be specified only if you would like to override their default values as defined in Section 4.1. Figure 1 shows the minimal recommended configuration file header; Figure 2 shows a more complex set of options.

```
TrustRootPath = c:\Trustroots
LogLevel      = 1
LogPath       = c:\WebSecurity\logs\WCgeneral.log
```

Figure 1: A small configuration file header. In this example, the trustroot store is identified and the logging functionality configured.

```
TrustRootPath = c:\WebSecurity\WCTrustroots
LogLevel      = 1
LogPath       = c:\WebSecurity\logs\WCgeneral.log
CacheEntries  = 500
MaxCacheAge   = 3600
ErrorDocument = c:\cTest\validation\custom_failed.html
PermittedSubtree = o=State Polytechnic Institute, c=US
BlacklistedServer = server1.example.com
```

Figure 2: A larger configuration file header. In this example the cache settings are also customized, as well as a custom error document specified. Also, access to the site is by default limited to certificates from State Polytechnic Institute, although this can be overridden at the individual directory level in the body of the configuration file.

5.2 Directory Options: The Configuration File Body

The body of the configuration file is composed of blocks of options for each directory tree, one per directory tree. Each block is identified by an initial line identifying the directory in question. There are no required options for the body of the configuration file, other than this leading identifier.

```
[/]
LDAP = ldap.spi.edu;389;
PermittedSubtree = o=State Polytechnic Institute, c=US
RequireFreshCRL  = Yes

[/Cog_Sci]
PermittedSubtree = ou=Computer Science, o=State College, c=US
PermittedSubtree = ou=Psychology, o=State College, c=US
PermittedSubtree = ou=Linguistics, o=State College, c=US
PermittedSubtree = ou=Philosophy, o=State College, c=US

[/Faculty]
ExcludedSubtree  = ou=Students, o=State College, c=US
ExcludedSubtree  = ou=Staff, o=State College, c=US
PermittedSubtree = cn=Jane Admin, ou=Staff, o=State College, c=US
PermittedSubtree = cn=Jose Admin, ou=Staff, o=State College, c=US

[/Grades]
AllowedPolicy      = 2.3.4.5.1
AllowedPolicy      = 2.3.4.5.4
RequireAllPolicies = Yes
ExtendedKeyUsage   = 1.2.3.4.5.6

[/HR]
MinKeySize         = 2048
RequireRecentCRL   = Yes
CRLFreshness       = 3600
```

Figure 3: A sample configuration file body, using various restrictions for the root, Cog_Sci, Grades, Faculty, and HR directories.

Note that in Figure 3, the restrictions specified in the first entry (that of the root directory) will be applied universally to all web server documents, as the entire web file system is in the subtree of the root. We see that the `PermittedSubtree` restrictions for the `Cog_Sci` subdirectory provide further name restrictions than those defined for the root directory. For `Faculty`, the `PermittedSubTree` option is used with `ExcludedSubtree` to provide small exceptions for administrative access to restricted files. Due to the sensitive nature of the documents, access to `Grades` is restricted based on certificate policy and the key usage policy. Similarly, the entire cert chain must have keys of size at least 2048 in order to access `HR`. For this last directory, Webcullis is also required to use CRLs that are no older than an hour in performing validation.

5.3 Configuration Scenarios

5.3.1 Restricting access to a particular department

Imagine an intranet server containing both information of general organizational interest and information (such as, for example, unreleased financial performance data) that should not be generally available. Webcullis makes it easy to quickly restrict access to such data using name constraints.

```
TrustRootPath = c:\WebSecurity\WCTrustroots
LogLevel      = 1
LogPath       = c:\WebSecurity\logs\WCgeneral.log
CacheEntries  = 500
MaxCacheAge   = 3600
ErrorDocument = c:\website\validation\custom_failed.html


[/]
LDAP = ldap.ketogen.com;389;
PermittedSubtree = o=Ketogen Pharmaceuticals, c=US
RequireFreshCRL  = Yes

[/Accounting]
PermittedSubtree = ou=Accounting, o=Ketogen Pharmaceuticals, c=US
```

Figure 4: Simple configuration file for an intranet server

5.3.2 Restricting access to particular individuals

Webcullis can also be used as a quick, simple form of access control list. This configuration file restricts an upcoming annual report only to those who are working on it.

```
TrustRootPath = c:\WebSecurity\WCTrustroots
LogLevel      = 1
LogPath       = c:\WebSecurity\logs\WCgeneral.log
CacheEntries  = 500
MaxCacheAge   = 3600
ErrorDocument = c:\website\validation\custom_failed.html


[/]
LDAP = ldap.ketogen.com;389;
PermittedSubtree = o=Ketogen Pharmaceuticals, c=US
RequireFreshCRL  = Yes

[/Accounting]
PermittedSubtree = ou=Accounting, o=Ketogen Pharmaceuticals, c=US

[/Accounting/FY06]
PermittedSubtree = CN=Joe Dimaggio, ou=Accounting, o=Ketogen Pharmaceuticals, c=US
PermittedSubtree = CN=Mickey Mantle, ou=Accounting, o=Ketogen Pharmaceuticals, c=US
PermittedSubtree = CN=Ted Williams, ou=Accounting, o=Ketogen Pharmaceuticals, c=US
```

Figure 5: Simple ACL

5.3.3 Excluding Low-Assurance certificates

Policy-based access control allows you to trust certificates issued under one policy but not another. Suppose that Ketogen's CA issues 3 policies. 2.16.840.1.101.9.8.7.1 is asserted on software-protected certificates, 2.16.840.1.101.9.8.7.2 is asserted on certifi-

cates protected by traditional hardware tokens, and 2.16.840.1.101.9.8.7.3 is asserted on certificates protected by biometric hardware tokens.

```
TrustRootPath = c:\WebSecurity\WCTrustroots
LogLevel      = 1
LogPath       = c:\WebSecurity\logs\WCgeneral.log
CacheEntries  = 500
MaxCacheAge   = 3600
ErrorDocument = c:\website\validation\custom_failed.html
# These should be set to "Yes" in most cases where
# policy-based restriction is in use
InitialExplicitPloicy = Yes
InhibitAnyPolicy = Yes


[/]
LDAP = ldap.ketogen.com;389;
PermittedSubtree = o=Ketogen Pharmaceuticals, c=US
RequireFreshCRL  = Yes
# General content is open to anyone with a software,
# hardware, or biometric hardware cert
AllowedPolicy = 2.16.840.1.101.9.8.7.1
AllowedPolicy = 2.16.840.1.101.9.8.7.2
AllowedPolicy = 2.16.840.1.101.9.8.7.3

[/Accounting]
PermittedSubtree = ou=Accounting, o=Ketogen Pharmaceuticals, c=US
# Require either hardware or hardware/biometric tokens for the
# accounting department
AllowedPolicy = 2.16.840.1.101.9.8.7.2
AllowedPolicy = 2.16.840.1.101.9.8.7.3
```

Figure 6: Policy Sample

5.3.4 Excluding inappropriately used keys

If you know that a particular CA uses the extended key usage extension, you can configure Webcullis to enforce it.

```
TrustRootPath = c:\WebSecurity\WCTrustroots
LogLevel      = 1
LogPath       = c:\WebSecurity\logs\WCgeneral.log
CacheEntries  = 500
MaxCacheAge   = 3600
ErrorDocument = c:\website\validation\custom_failed.html



[/]
LDAP = ldap.ketogen.com;389;
PermittedSubtree = o=Ketogen Pharmaceuticals, c=US
RequireFreshCRL  = Yes
# require the "TLS Client Auth" extended key usage extension be present,
# since this CA always sets it.
ExtendedKeyUsage = 1.3.6.1.5.5.7.3.2


[/Accounting]
PermittedSubtree = ou=Accounting, o=Ketogen Pharmaceuticals, c=US
```

Figure 7: ExtendedKeyUsage Sample

5.3.5 Key Size Restriction

If an organization is in the process of transitioning end entities from one key size to another, it may be desirable to restrict access to some resources based on key size.

```
TrustRootPath = c:\WebSecurity\WCTrustroots
LogLevel      = 1
LogPath       = c:\WebSecurity\logs\WCgeneral.log
CacheEntries  = 500
MaxCacheAge   = 3600
ErrorDocument = c:\website\validation\custom_failed.html



[/]
LDAP = ldap.ketogen.com;389;
PermittedSubtree = o=Ketogen Pharmaceuticals, c=US
RequireFreshCRL  = Yes

[/Accounting]
PermittedSubtree = ou=Accounting, o=Ketogen Pharmaceuticals, c=US

[/Archiving]
# This is a long-term archiving application.
# Require at least a 2048-bit RSA key for entry.
MinKeySize = 2048
```

Figure 8: MinKeySize Sample

5.3.6 Prohibit test certificates from being used in production

This server is generally available to anyone in the organization with a valid, non-test certificate. This CA issues 3 production policies. 2.16.840.1.101.9.8.7.1 is asserted on software-protected certificates, 2.16.840.1.101.9.8.7.2 is asserted on certificates protected by traditional hardware tokens, and 2.16.840.1.101.9.8.7.3 is asserted on certificates protected by biometric hardware tokens. For test certificates, they assert the policy 2.16.840.1.101.9.8.7.6.

```
TrustRootPath = c:\WebSecurity\WCTrustroots
LogLevel      = 1
LogPath       = c:\WebSecurity\logs\WCgeneral.log
CacheEntries  = 500
MaxCacheAge   = 3600
ErrorDocument = c:\website\validation\custom_failed.html
# These should be set to "Yes" in most cases where policy-based restriction is in use
InitialExplicitPloicy = Yes
InhibitAnyPolicy = Yes



[/]
LDAP = ldap.ketogen.com;389;
PermittedSubtree = o=Ketogen Pharmaceuticals, c=US
RequireFreshCRL  = Yes
AllowedPolicy = 2.16.840.1.101.9.8.7.1
AllowedPolicy = 2.16.840.1.101.9.8.7.2
AllowedPolicy = 2.16.840.1.101.9.8.7.3
```

Figure 9: Another Policy Sample

6 For More Information

This document contains the information necessary to install and configure the Webcullis plug-in to the IIS web server. We suggest the following resources should you need more information to help you during this process:

- The IIS documentation http://www.microsoft.com/windows2000/en/server/iis/
- " Guide to the Secure Configuration of Microsoft Internet and Information Services," http://www.nsa.gov/snac/downloads_miis.cfm?MenuID=scg10.3.1.4
- "RFC 3280: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List Profile," http://www.faqs.org/rfcs/rfc3280.html