Center for Internet Security®

CIS Microsoft IIS 10 Benchmark

v1.0.0 - 03-31-2017

Table of Contents

Overview

This document, CIS Microsoft IIS 10 Benchmark, provides prescriptive guidance for establishing a secure configuration posture for Microsoft IIS 10. This guide was tested against Microsoft IIS 10 running on Microsoft Windows Server 2016. To obtain the latest version of this guide, please visit <http://benchmarks.cisecurity.org>. If you have questions, comments, or have identified ways to improve this guide, please write us at feedback@cisecurity.org.

Intended Audience

This document is intended for system and application administrators, security specialists, auditors, help desk, and platform deployment personnel who plan to develop, deploy, assess, or secure solutions that incorporate Microsoft IIS 10.

Consensus Guidance

This benchmark was created using a consensus review process comprised of subject matter experts. Consensus participants provide perspective from a diverse set of backgrounds including consulting, software development, audit and compliance, security research, operations, government, and legal.

Each CIS benchmark undergoes two phases of consensus review. The first phase occurs during initial benchmark development. During this phase, subject matter experts convene to discuss, create, and test working drafts of the benchmark. This discussion occurs until consensus has been reached on benchmark recommendations. The second phase begins after the benchmark has been published. During this phase, all feedback provided by the Internet community is reviewed by the consensus team for incorporation in the benchmark. If you are interested in participating in the consensus process, please visit https://community.cisecurity.org.

Typographical Conventions

The following typographical conventions are used throughout this guide:

Convention	Meaning
`Stylized Monospace font`	Used for blocks of code, command, and script examples. Text should be interpreted exactly as presented.
Monospace font	Used for inline code, commands, or examples. Text should be interpreted exactly as presented.
<italic font in brackets>	Italic texts set in angle brackets denote a variable requiring substitution for a real value.
Italic font	Used to denote the title of a book, article, or other publication.
Note	Additional information or caveats

Scoring Information

A scoring status indicates whether compliance with the given recommendation impacts the assessed target's benchmark score. The following scoring statuses are used in this benchmark:

Scored

Failure to comply with "Scored" recommendations will decrease the final benchmark score. Compliance with "Scored" recommendations will increase the final benchmark score.

Not Scored

Failure to comply with "Not Scored" recommendations will not decrease the final benchmark score. Compliance with "Not Scored" recommendations will not increase the final benchmark score.

Profile Definitions

The following configuration profiles are defined by this Benchmark:

- **Level 1 - IIS 10**

  Items in this profile apply to Microsoft IIS 10 running on Microsoft Windows Server 2016 and intend to:

  - be practical and prudent;
  - provide a clear security benefit; and
  - not inhibit the utility of the technology beyond acceptable means.

- **Level 2 - IIS 10**

  This profile extends the "Level 1 - IIS 10" profile. Items in this profile apply to Microsoft IIS 10 running on Microsoft Windows Server 2016 and exhibit one or more of the following characteristics:

  - are intended for environments or use cases where security is paramount
  - acts as defense in depth measure
  - may negatively inhibit the utility or performance of the technology.

Acknowledgements

This benchmark exemplifies the great things a community of users, vendors, and subject matter experts can accomplish through consensus collaboration. The CIS community thanks the entire consensus team with special recognition to the following individuals who contributed greatly to the creation of this guide:

Contributor
Jordan Rakoske
Philippe Langlois

Editor
Terri Donahue
Victor Dzheyranov

The Center for Internet Security extends special recognition and thanks to Microsoft IIS team for their collaboration and validation of this guide.

Recommendations

1 Basic Configurations

This section contains basic Web server-level recommendations.

1.1 Ensure web content is on non-system partition (Scored)

Profile Applicability:

- Level 1 - IIS 10

Description:

Web resources published through IIS are mapped, via Virtual Directories, to physical locations on disk. It is recommended to map all Virtual Directories to a non-system disk volume.

Rationale:

Isolating web content from system files may reduce the probability of:

- Web sites/applications exhausting system disk space
- File IO vulnerability in the web site/application from affecting the confidentiality and/or integrity of system files

Audit:

Execute the following command to ensure no virtual directories are mapped to the system drive:

```
%systemroot%\system32\inetsrv\appcmd list vdir
```

Remediation:

1. Browse to web content in `C:\\inetpub\\wwwroot\\`
2. Copy or cut content onto a dedicated and restricted web folder on a non-system drive such as `D:\\webroot\\`
3. Change mappings for any applications or Virtual Directories to reflect the new location

To change the mapping for the application named app1 which resides under the Default Web Site, open IIS Manager:

1. Expand the server node
2. Expand Sites
3. Expand Default Web Site
4. Click on app1
5. In the Actions pane, select Basic Settings
6. In the Physical path text box, put the new location of the application, `D:\\wwwroot\\app1` in the example above

Default Value:

The default location for web content is: %systemdrive%\inetpub\wwwroot.

References:

1. http://blogs.iis.net/thomad/archive/2008/02/10/moving-the-iis7-inetpub-directory-to-a-different-drive.aspx

CIS Controls:

14 Controlled Access Based on the Need to Know
Controlled Access Based on the Need to Know

1.2 Ensure 'host headers' are on all sites (Scored)

Profile Applicability:

- Level 1 - IIS 10

Description:

Host headers provide the ability to host multiple websites on the same IP address and port. It is recommended that host headers be configured for all sites. Wildcard host headers are now supported.

Rationale:

Requiring a Host header for all sites may reduce the probability of:

- DNS rebinding attacks successfully compromising or abusing site data or functionality
- IP-based scans successfully identifying or interacting with a target application hosted on IIS

Note: If a wildcard DNS entry exists and a wildcard host header is used, you may be serving data to more domains than intended.

Audit:

Execute the following command to identify sites that are not configured to require host headers:

```
%systemroot%\system32\inetsrv\appcmd list sites
```

All sites will be listed as such: `SITE "Default Web Site" (id:1,bindings:http/\*:80:test.com,state:Started) SITE "badsite" (id:3,bindings:http/\*:80:,state:Started)` For all non-SSL sites, ensure that the *IP:port:host* binding triplet contains a host name. In the example above, the first site is configured as recommended given the `Default Web Site` has a host header of `test.com`. `badsite`, however, does not have a host header configured - it shows `\*:80:` which means all IPs over port 80, with no host header.

Remediation:

Obtain a listing of all sites by using the following `appcmd.exe` command:

```
%systemroot%\system32\inetsrv\appcmd list sites
```

Perform the following in IIS Manager to configure host headers for the Default Web Site:

1. Open IIS Manager
2. In the Connections pane expand the Sites node and select Default Web Site
3. In the Actions pane click Bindings
4. In the Site Bindings dialog box, select the binding for which host headers are going to be configured, Port 80 in this example
5. Click Edit
6. Under host name, enter the sites FQDN, such as *<www.examplesite.com>*
7. Click OK, then Close

Note: Requiring a host header may impair site functionality for HTTP/1.0 clients.

Default Value:

By default, host headers are not required or set up automatically.

References:

1. http://technet.microsoft.com/en-us/library/cc753195%28WS.10%29.aspx
2. http://crypto.stanford.edu/dns/dns-rebinding.pdf
3. http://www.sslshopper.com/article-ssl-host-headers-in-iis-7.html
4. http://blogs.iis.net/thomad/archive/2008/01/25/ssl-certificates-on-sites-with-host-headers.aspx
5. https://www.iis.net/learn/get-started/whats-new-in-iis-10/wildcard-host-header-support

CIS Controls:

18 Application Software Security
Application Software Security

1.3 Ensure 'directory browsing' is set to disabled (Scored)

Profile Applicability:

- Level 1 - IIS 10

Description:

Directory browsing allows the contents of a directory to be displayed upon request from a web client. If directory browsing is enabled for a directory in Internet Information Services, users receive a page that lists the contents of the directory when the following two conditions are met:

1. No specific file is requested in the URL
2. The Default Documents feature is disabled in IIS, or if it is enabled, IIS is unable to locate a file in the directory that matches a name specified in the IIS default document list

It is recommended that directory browsing be disabled.

Rationale:

Ensuring that directory browsing is disabled may reduce the probability of disclosing sensitive content that is inadvertently accessible via IIS.

Audit:

Perform the following to verify that Directory Browsing has been disabled at the server level:

```
%systemroot%\system32\inetsrv\appcmd list config /section:directoryBrowse
```

If the server is configured as recommended, the following will be displayed:

```
<system.webServer>
  <directoryBrowse enabled="false" />
<system.webServer>
```

Remediation:

Directory Browsing can be set by using the UI, running `appcmd.exe` commands, by editing configuration files directly, or by writing WMI scripts. To disable directory browsing at the server level using an `appcmd.exe` command:

```
%systemroot%\system32\inetsrv\appcmd set config /section:directoryBrowse
/enabled:false
```

Default Value:

In IIS, directory browsing is disabled by default.

References:

1. http://technet.microsoft.com/en-us/library/cc725840%28WS.10%29.aspx
2. http://technet.microsoft.com/en-us/library/cc731109%28WS.10%29.aspx

CIS Controls:

18 Application Software Security
Application Software Security

1.4 Ensure 'application pool identity' is configured for all application pools (Scored)

Profile Applicability:

- Level 1 - IIS 10

Description:

Application Pool Identities are the actual users/authorities that will run the worker process - `w3wp.exe`. Assigning the correct user authority will help ensure that applications can function properly, while not giving overly permissive permissions on the system. These identities can further be used in ACLs to protect system content. It is recommended that each Application Pool run under a unique identity.

IIS has additional built-in least privilege identities intended for use by Application Pools. It is recommended that the default Application Pool Identity be changed to a least privilege principle other than Network Service. Furthermore, it is recommended that all application pool identities be assigned a unique least privilege principal.

To achieve isolation in IIS, application pools can be run as separate identities. IIS can be configured to automatically use the application pool identity if no anonymous user account is configured for a Web site. This can greatly reduce the number of accounts needed for Web sites and make management of the accounts easier. It is recommended the Application Pool Identity be set as the Anonymous User Identity.

The name of the Application Pool account corresponds to the name of the Application Pool. Application Pool Identities were introduced in Windows Server 2008 SP2. It is recommended that Application Pools be set to run as `ApplicationPoolIdentity` unless there is an underlying reason that the application pool needs to run as a specified end user account. One example where this is needed is for web farms using Kerberos authentication.

Rationale:

Setting Application Pools to use unique least privilege identities such as `ApplicationPoolIdentity` reduces the potential harm the identity could cause should the application ever become compromised.

Additionally, it will simplify application pools configuration and account management.

Audit:

To verify the Application Pools have been set to run under the ApplicationPoolIdentity using IIS Manager:

1. Open IIS Manager
2. Open the Application Pools node underneath the machine node; select Application Pool to be verified
3. Right click the Application Pool and select Advanced Settings…
4. Under the Process Model section, locate the Identity option and ensure that `ApplicationPoolIdentity` is set

This configuration is stored in the same `applicationHost.config` file for web sites and application/virtual directories, at the bottom of the file, surrounded by `<location path="path/to/resource">` tags.

To verify that any new Application Pools use the ApplicationPoolIdentity, execute the following command to determine if the Application Pool default has been changed to `ApplicationPoolIdentity`:

```
%systemroot%\system32\inetsrv\appcmd list config /section:applicationPools
```

Remediation:

The default Application Pool identity may be set for an application using the IIS Manager GUI, using `AppCmd.exe` commands in a command-line window, directly editing the configuration files, or by writing WMI scripts. Perform the following to change the default identity to the built-in `ApplicationPoolIdentity` in the IIS Manager GUI:

1. Open the IIS Manager GUI
2. In the connections pane, expand the server node and click Application Pools
3. On the Application Pools page, select the `DefaultAppPool`, and then click Advanced Settings in the Actions pane
4. For the Identity property, click the `'...'` button to open the Application Pool Identity dialog box
5. Select the Built-in account option choose `ApplicationPoolIdentity` from the list, or input a unique application user created for this purpose
6. Restart IIS

To change the `ApplicationPool` identity to the built-in `ApplicationPoolIdentity` using `AppCmd.exe`, run the following from a command prompt:

```
%systemroot%\system32\inetsrv\appcmd set config /section:applicationPools
/[name='<your apppool>'].processModel.identityType:ApplicationPoolIdentity
```

The example code above will set just the `DefaultAppPool`. Run this command for each configured Application Pool. Additionally, `ApplicationPoolIdentity` can be made the default for all Application Pools by using the Set Application Pool Defaults action on the Application Pools node.

If using a custom defined Windows user such as a dedicated service account, that user will need to be a member of the IIS_IUSRS group. The IIS_IUSRS group has access to all the necessary file and system resources so that an account, when added to this group, can seamlessly act as an application pool identity.

Default Value:

By Default, the `DefaultAppPool` in IIS is configured to use the `ApplicationPoolIdentity` account.

References:

1. http://technet.microsoft.com/en-us/library/cc771170%28WS.10%29.aspx
2. http://learn.iis.net/page.aspx/140/understanding-built-in-user-and-group-accounts-in-iis-7/
3. http://learn.iis.net/page.aspx/624/application-pool-identities/
4. http://blogs.iis.net/tomwoolums/archive/2008/12/17/iis-7-0-application-pools.aspx

CIS Controls:

18 Application Software Security
Application Software Security

1.5 Ensure 'unique application pools' is set for sites (Scored)

Profile Applicability:

- Level 1 - IIS 10

Description:

IIS introduced a new security feature called Application Pool Identities that allows Application Pools to be run under unique accounts without the need to create and manage local or domain accounts. It is recommended that all Sites run under unique, dedicated Application Pools.

Rationale:

By setting sites to run under unique Application Pools, resource-intensive applications can be assigned to their own application pools which could improve server and application performance.In addition, it can help maintain application availability: if an application in one pool fails, applications in other pools are not affected.Last, isolating applications helps mitigate the potential risk of one application being allowed access to the resources of another application. It is also recommended to stop any application pool that is not in use or was created by an installation such as .Net 4.0.

Audit:

The following `appcmd.exe` command will give a listing of all applications configured, which site they are in, which application pool is serving them and which application pool identity they are running under:

```
%systemroot%\system32\inetsrv\appcmd list app
```

The output of this command will be similar to the following: `APP "Default Web Site/" (applicationPool:DefaultAppPool)`

1. Run the above command and ensure a unique application pool is assigned for each site listed

Remediation:

1. Open IIS Manager
2. Open the Sites node underneath the machine node
3. Select the Site to be changed
4. In the Actions pane, select Basic Settings
5. Click the Select… box next to the Application Pool text box
6. Select the desired Application Pool

7. Once selected, click OK

Default Value:

By default, all Sites created will use the Default Application Pool (`DefaultAppPool`).

References:

1. http://technet.microsoft.com/en-us/library/cc753449%28WS.10%29.aspx
2. http://blogs.iis.net/tomwoolums/archive/2008/12/17/iis-7-0-application-pools.aspx
3. http://learn.iis.net/page.aspx/624/application-pool-identities/

CIS Controls:

14.4 Protect Information With Access Control Lists
All information stored on systems shall be protected with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principal that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

1.6 Ensure 'application pool identity' is configured for anonymous user identity (Scored)

Profile Applicability:

- Level 1 - IIS 10

Description:

To achieve isolation in IIS, application pools can be run as separate identities. IIS can be configured to automatically use the application pool identity if no anonymous user account is configured for a Web site. This can greatly reduce the number of accounts needed for Web sites and make management of the accounts easier. It is recommended the Application Pool Identity be set as the Anonymous User Identity.

Rationale:

Configuring the anonymous user identity to use the application pool identity will help ensure site isolation - provided sites are set to use the application pool identity. Since a unique principal will run each application pool, it will ensure the identity is least privilege. Additionally, it will simplify Site management.

Audit:

Find and open the `applicationHost.config` file and verify that the `userName` attribute of the `anonymousAuthentication` tag is set to a blank string:

```
<system.webServer>
  <security>
    <authentication>
      <anonymousAuthentication userName="" />
    </authentication>
  </security>
</system.webServer>
```

This configuration is stored in the same `applicationHost.config` file for web sites and application/virtual directories, at the bottom of the file, surrounded by `<location path="path/to/resource">` tags.

Remediation:

The Anonymous User Identity can be set to Application Pool Identity by using the IIS Manager GUI, using `AppCmd.exe` commands in a command-line window, directly editing the configuration files, or by writing WMI scripts. Perform the following to set the username attribute of the `anonymousAuthentication` node in the IIS Manager GUI:

1. Open the IIS Manager GUI and navigate to the desired server, site, or application
2. In Features View, find and double-click the Authentication icon
3. Select the Anonymous Authentication option and in the Actions pane select Edit...
4. Choose Application pool identity in the modal window and then press the OK button

To use `AppCmd.exe` to configure `anonymousAuthentication` at the server level, the command would look like this:

```
%systemroot%\system32\inetsrv\appcmd set config -
section:anonymousAuthentication /username:"" --password
```

Default Value:

The default identity for the anonymous user is the IUSR virtual account.

References:

1. http://learn.iis.net/page.aspx/202/application-pool-identity-as-anonymous-user/
2. http://learn.iis.net/page.aspx/624/application-pool-identities/

CIS Controls:

14.4 Protect Information With Access Control Lists
All information stored on systems shall be protected with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principal that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

2 Configure Authentication and Authorization

This section contains recommendations around the different layers of authentication in IIS.

2.1 Ensure 'global authorization rule' is set to restrict access (Not Scored)

Profile Applicability:

- Level 1 - IIS 10

Description:

IIS introduced URL Authorization, which allows the addition of Authorization rules to the actual URL, instead of the underlying file system resource, as a way to protect it. Authorization rules can be configured at the server, web site, folder (including Virtual Directories), or file level. The native URL Authorization module applies to all requests, whether they are .NET managed or other types of files (e.g. static files or ASP files). It is recommended that URL Authorization be configured to only grant access to the necessary security principals.

Rationale:

Configuring a global Authorization rule that restricts access will ensure inheritance of the settings down through the hierarchy of web directories; if that content is copied elsewhere, the authorization rules flow with it. This will ensure access to current and future content is only granted to the appropriate principals, mitigating risk of accidental or unauthorized access.

Audit:

At the web site or application level, verify that the authorization rule configured has been applied:

1. Connect to Internet Information Services (IIS Manager)
2. Select the site or application where Authorization was configured
3. Select Authorization Rules and verify the configured rules were added

To verify an authorization rule specifying no access to all users except the `Administrators` group, browse to and open the `web.config` file for the configured site/application/content:

```
<configuration>
  <system.webServer>
    <security>
      <authorization>
```

```
        <remove users="*" roles="" verbs="" />
        <add accessType="Allow" roles="administrators" />
      </authorization>
    </security>
  </system.webServer>
</configuration>
```

Remediation:

To configure URL Authorization at the server level using IIS Manager:

1. Connect to Internet Information Services (IIS Manager)
2. Select the server
3. Select Authorization Rules
4. Remove the "Allow All Users" rule
5. Click Add Allow Rule…
6. Allow access to the user(s), user groups, or roles that are authorized across all of the web sites and applications (e.g. the Administrators group)

Default Value:

The default server-level setting is to allow all users access.

References:

1. http://www.iis.net/learn/manage/configuring-security/understanding-iis-url-authorization
2. http://www.iis.net/learn/get-started/whats-new-in-iis-7/changes-in-security-between-iis-60-and-iis-7-and-above#Authorization

CIS Controls:

18 Application Software Security
Application Software Security

2.2 Ensure access to sensitive site features is restricted to authenticated principals only (Not Scored)

Profile Applicability:

- Level 1 - IIS 10

Description:

IIS supports both challenge-based and login redirection-based authentication methods. Challenge-based authentication methods, such as Integrated Windows Authentication, require a client to respond correctly to a server-initiated challenge. A login redirection-based authentication method such as Forms Authentication relies on redirection to a login page to determine the identity of the principal. Challenge-based authentication and login redirection-based authentication methods cannot be used in conjunction with one another.

Public servers/sites are typically configured to use Anonymous Authentication. This method typically works, provided the content or services is intended for use by the public. When sites, applications, or specific content containers are not intended for anonymous public use, an appropriate authentication mechanism should be utilized. Authentication will help confirm the identity of clients who request access to sites, application, and content. IIS provides the following authentication modules by default:

- Anonymous Authentication - allows anonymous users to access sites, applications, and/or content
- Integrated Windows Authentication - authenticates users using the NTLM or Kerberos protocols; Kerberos v5 requires a connection to Active Directory
- ASP.NET Impersonation - allows ASP.NET applications to run under a security context different from the default security context for an application
- Forms Authentication - enables a user to login to the configured space with a valid user name and password which is then validated against a database or other credentials store
- Basic authentication - requires a valid user name and password to access content
- Client Certificate Mapping Authentication - allows automatic authentication of users who log on with client certificates that have been configured; requires SSL
- Digest Authentication - uses Windows domain controller to authenticate users who request access

Note that none of the challenge-based authentication modules can be used at the same time Forms Authentication is enabled for certain applications/content. Forms Authentication does not rely on IIS authentication, so anonymous access for the ASP.NET application can be configured if Forms Authentication will be used.

It is recommended that sites containing sensitive information, confidential data, or non-public web services be configured with a credentials-based authentication mechanism.

Rationale:

Configuring authentication will help mitigate the risk of unauthorized users accessing data and/or services, and in some cases reduce the potential harm that can be done to a system.

Audit:

To verify that the authentication module is enabled for a specific site, application, or content, browse to and open the `web.config` file pertaining to the content. Verify the configuration file now has a mode defined within the `<authentication>` tags. The example below shows that Forms Authentication is configured, cookies will always be used, and SSL is required:

```
<system.web>
  <authentication>
    <forms cookieless="UseCookies" requireSSL="true" />
  </authentication>
</system.web>
```

Remediation:

When configuring an authentication module for the first time, each mechanism must be completely configured before use.

Enabling authentication can be performed by using the user interface (UI), running `AppCmd.exe` commands in a command-line window, editing configuration files directly, or by writing WMI scripts. To verify an authentication mechanism is in place for sensitive content using the IIS Manager GUI:

1. Open IIS Manager and navigate to level with sensitive content
2. In Features View, double-click Authentication
3. On the Authentication page, make sure an authentication module is enabled, while anonymous authentication is enabled (Forms Authentication can have anonymous as well)
4. If necessary, select the desired authentication module, then in the Actions pane, click Enable

Default Value:

The default installation of IIS supports Anonymous Authentication without further electing additional methods.

References:

1. http://learn.iis.net/page.aspx/377/using-aspnet-forms-authentication/rev/1
2. http://learn.iis.net/page.aspx/244/how-to-take-advantage-of-the-iis7-integrated-pipeline/
3. http://technet.microsoft.com/en-us/library/cc733010%28WS.10%29.aspx
4. http://msdn.microsoft.com/en-us/library/aa480476.aspx
5. https://technet.microsoft.com/en-us/library/hh831496(v=ws.11).aspx

CIS Controls:

14.4 Protect Information With Access Control Lists

All information stored on systems shall be protected with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principal that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

2.3 Ensure 'forms authentication' require SSL (Scored)

Profile Applicability:

- Level 1 - IIS 10

Description:

Forms-based authentication can pass credentials across the network in clear text. It is therefore imperative that the traffic between client and server be encrypted using SSL, especially in cases where the site is publicly accessible. It is recommended that communications with any portion of a site using Forms Authentication be encrypted using SSL.

NOTE Due to identified security vulnerabilities, SSL is no longer considered to provides adequate protection for a sensitive information.

Rationale:

Requiring SSL for Forms Authentication will protect the confidentiality of credentials during the login process, helping mitigate the risk of stolen user information.

Audit:

To verify that SSL is required for forms authentication for a specific site, application, or content, browse to and open the `web.config` file for the level in which forms authentication was enabled. Verify the tag `<forms requireSSL="true" />`:

```
<system.web>
  <authentication>
    <forms requireSSL="true" />
  </authentication>
</system.web>
```

Remediation:

1. Open IIS Manager and navigate to the appropriate tier
2. In Features View, double-click Authentication
3. On the Authentication page, select Forms Authentication
4. In the Actions pane, click Edit
5. Check the Requires SSL checkbox in the cookie settings section, click OK

Default Value:

SSL is not required when Forms Authentication is enabled.

References:

1. http://technet.microsoft.com/en-us/library/cc771077(WS.10).aspx

CIS Controls:

14.2 Encrypt All Sensitive Information Over Less-trusted Networks
All communication of sensitive information over less-trusted networks should be encrypted. Whenever information flows over a network with a lower trust level, the information should be encrypted.

2.4 Ensure 'forms authentication' is set to use cookies (Scored)

Profile Applicability:

- Level 2 - IIS 10

Description:

Forms Authentication can be configured to maintain the site visitor's session identifier in either a URI or cookie. It is recommended that Forms Authentication be set to use cookies.

Rationale:

Using cookies to manage session state may help mitigate the risk of session hi-jacking attempts by preventing ASP.NET from having to move session information to the URL. Moving session information identifiers into the URL may cause session IDs to show up in proxy logs, browsing history, and be accessible to client scripting via `document.location`.

Audit:

Locate and open the `web.config` for the configured application. Verify the presence of `<forms cookieless="UseCookies" />`.

```
<system.web>
  <authentication>
    <forms cookieless="UseCookies" requireSSL="true" timeout="30" />
  </authentication>
</system.web>
```

Remediation:

1. Open IIS Manager and navigate to the level where Forms Authentication is enabled
2. In Features View, double-click Authentication
3. On the Authentication page, select Forms Authentication
4. In the Actions pane, click Edit
5. In the Cookie settings section, select Use cookies from the Mode dropdown

Default Value:

The default setting for Cookie Mode is Auto Detect which will only use cookies if the device profile supports cookies.

References:

1. http://technet.microsoft.com/en-us/library/cc732830%28WS.10%29.aspx

CIS Controls:

18 Application Software Security
Application Software Security

2.5 Ensure 'cookie protection mode' is configured for forms authentication (Scored)

Profile Applicability:

- Level 1 - IIS 10

Description:

The cookie protection mode defines the protection Forms Authentication cookies will be given within a configured application. The four cookie protection modes that can be defined are:

- Encryption and validation - Specifies that the application use both data validation and encryption to help protect the cookie; this option uses the configured data validation algorithm (based on the machine key) and triple-DES (3DES) for encryption, if available and if the key is long enough (48 bytes or more)
- None - Specifies that both encryption and validation are disabled for sites that are using cookies only for personalization and have weaker security requirements
- Encryption - Specifies that the cookie is encrypted by using Triple-DES or DES, but data validation is not performed on the cookie; cookies used in this manner might be subject to plain text attacks
- Validation - Specifies that a validation scheme verifies that the contents of an encrypted cookie have not been changed in transit

It is recommended that cookie protection mode always encrypt and validate Forms Authentication cookies.

Rationale:

By encrypting and validating the cookie, the confidentiality and integrity of data within the cookie is assured. This helps mitigate the risk of attacks such as session hijacking and impersonation.

Audit:

Locate and open the `web.config` for the configured application. Verify the presence of `<forms protection="All" />`.

```
<system.web>
  <authentication>
    <forms cookieless="UseCookies" protection="All" />
  </authentication>
</system.web>
```

The `protection="All"` property will only show up if cookie protection mode was set to something different, and then changed to Encryption and validation. To truly verify the `protection="All"` property in the `web.config`, the protection mode can be changed, and then changed back. Conversely, the `protection="All"` line can be added to the `web.config` manually.

Remediation:

Cookie protection mode can be configured by using the user interface (UI), by running `Appcmd.exe` commands in a command-line window, by editing configuration files directly, or by writing WMI scripts. Using IIS Manager:

1. Open IIS Manager and navigate to the level where Forms Authentication is enabled
2. In Features View, double-click Authentication
3. On the Authentication page, select Forms Authentication
4. In the Actions pane, click Edit
5. In the Cookie settings section, verify the drop-down for Protection mode is set for Encryption and validation

Default Value:

When cookies are used for Forms Authentication, the default cookie protection mode is `All`, meaning the application encrypts and validates the cookie.

References:

1. http://technet.microsoft.com/en-us/library/cc731804%28WS.10%29.aspx

CIS Controls:

18 Application Software Security
Application Software Security

2.6 Ensure transport layer security for 'basic authentication' is configured (Scored)

Profile Applicability:

- Level 1 - IIS 10

Description:

Basic Authentication can pass credentials across the network in clear text. It is therefore imperative that the traffic between client and server be encrypted, especially in cases where the site is publicly accessible and is recommended that TLS be configured and required for any Site or Application using Basic Authentication.

Rationale:

Credentials sent in clear text can be easily intercepted by malicious code or persons. Enforcing the use of Transport Layer Security will help mitigate the chances of hijacked credentials.

Audit:

Once transport layer security has been configured and required for a Site or application, only the https:// address will be available. Attempt loading the Site or application for which Basic Authentication is configured using http://, the requests will fail and IIS will throw a 403.4 - Forbidden error.

Remediation:

To protect Basic Authentication with transport layer security:

1. Open IIS Manager
2. In the Connections pane on the left, select the server to be configured
3. In the Connections pane, expand the server, then expand Sites and select the site to be configured
4. In the Actions pane, click Bindings; the Site Bindings dialog appears
5. If an HTTPS binding is available, click Close and see below "To require SSL"
6. If no HTTPS binding is visible, perform the following steps

To add an HTTPS binding:

1. In the Site Bindings dialog, click Add; the Add Site Binding dialog appears
2. Under Type, select https
3. Under SSL certificate, select an X.509 certificate

4. Click OK, then close

To require SSL:

1. In Features View, double-click SSL Settings
2. On the SSL Settings page, select Require SSL.
3. In the Actions pane, click Apply

Default Value:

Transport Layer Security is not enabled by default when Basic Authentication is configured.

References:

1. http://technet.microsoft.com/en-us/library/dd378853%28WS.10%29.aspx

CIS Controls:

14.2 Encrypt All Sensitive Information Over Less-trusted Networks
All communication of sensitive information over less-trusted networks should be encrypted. Whenever information flows over a network with a lower trust level, the information should be encrypted.

2.7 Ensure 'passwordFormat' is not set to clear (Scored)

Profile Applicability:

- Level 1 - IIS 10

Description:

The `<credentials>` element of the `<authentication>` element allows optional definitions of name and password for IIS Manager User accounts within the configuration file. Forms based authentication also uses these elements to define the users. IIS Manager Users can use the administration interface to connect to sites and applications in which they've been granted authorization. Note that the `<credentials>` element only applies when the default provider, `ConfigurationAuthenticationProvider`, is configured as the authentication provider. It is recommended that `passwordFormat` be set to a value other than `Clear`, such as `SHA1`.

Rationale:

Authentication credentials should always be protected to reduce the risk of stolen authentication credentials.

Audit:

Locate and open the configuration file for the configured application. Verify the `credentials element` is not present:

```
<configuration>
  <system.web>
    <authentication mode="Forms">
      <forms name="SampleApp" loginUrl="/login.aspx">
        <credentials passwordFormat="SHA1">
          <user
            name="<em>UserName1</em>"
            password="<em>SHA1EncryptedPassword1</em>"/>
          <user
            name="<em>UserName2</em>"
            password="<em>SHA1EncryptedPassword2</em>"/>
        </credentials>
      </forms>
    </authentication>
  </system.web>
</configuration>
```

Remediation:

Authentication mode is configurable at the `machine.config`, root-level `web.config`, or application-level `web.config`:

1. Locate and open the configuration file where the credentials are stored
2. Find the `<credentials>` element
3. If present, ensure `passwordFormat` is not set to `Clear`
4. Change `passwordFormat`to `SHA1`

The clear text passwords will need to be replaced with the appropriate hashed version.

Default Value:

The default `passwordFormat`method is `SHA1`.

References:

1. http://www.iis.net/ConfigReference/system.webServer/management/authentication/credentials
2. http://msdn.microsoft.com/en-us/library/bb422401%28VS.90%29.aspx

CIS Controls:

16.14 Encrypt/Hash All Authentication Files And Monitor Their Access
Verify that all authentication files are encrypted or hashed and that these files cannot be accessed without root or administrator privileges. Audit all access to password files in the system.

2.8 Ensure 'credentials' are not stored in configuration files (Scored)

Profile Applicability:

- Level 2 - IIS 10

Description:

The `<credentials>` element of the `<authentication>` element allows optional definitions of name and password for IIS Manager User accounts within the configuration file. Forms based authentication also uses these elements to define the users. IIS Manager Users can use the administration interface to connect to sites and applications in which they've been granted authorization. Note that the `<credentials>` element only applies when the default provider, `ConfigurationAuthenticationProvider`, is configured as the authentication provider. It is recommended to avoid storing passwords in the configuration file even in form of hash.

Rationale:

Authentication credentials should always be protected to reduce the risk of stolen authentication credentials. For security reasons, it is recommended that user credentials not be stored an any IIS configuration files.

Audit:

Locate and open the configuration file for the configured application. Verify the `credentials element` is not present:

```
<configuration>
  <system.web>
    <authentication mode="Forms">
      <forms name="SampleApp" loginUrl="/login.aspx">
      </forms>
    </authentication>
  </system.web>
</configuration>
```

Remediation:

Authentication mode is configurable at the `machine.config`, root-level `web.config`, or application-level `web.config`:

1. Locate and open the configuration file where the credentials are stored
2. Find the `<credentials>` element
3. If present, remove the section

This will remove all references to stored users in the configuration files.

Default Value:

The default `passwordFormat` method is `SHA1`.

References:

1. http://www.iis.net/ConfigReference/system.webServer/management/authentication/credentials
2. http://msdn.microsoft.com/en-us/library/bb422401%28VS.90%29.aspx

CIS Controls:

16.14 Encrypt/Hash All Authentication Files And Monitor Their Access
Verify that all authentication files are encrypted or hashed and that these files cannot be accessed without root or administrator privileges. Audit all access to password files in the system.

3 ASP.NET Configuration Recommendations

This section contains recommendations specific to ASP.NET.

3.1 Ensure 'deployment method retail' is set (Scored)

Profile Applicability:

- Level 1 - IIS 10

Description:

The `<deployment retail>` switch is intended for use by production IIS servers. This switch is used to help applications run with the best possible performance and least possible security information leakages by disabling the application's ability to generate trace output on a page, disabling the ability to display detailed error messages to end users, and disabling the debug switch. Often times, switches and options that are developer-focused, such as failed request tracing and debugging, are enabled during active development. It is recommended that the deployment method on any production server be set to `retail`.

Rationale:

Utilizing the switch specifically intended for production IIS servers will eliminate the risk of vital application and system information leakages that would otherwise occur if tracing or debug were to be left enabled, or `customErrors` were to be left off.

Audit:

After the next time IIS is restarted, open the `machine.config` file and verify that `<deployment retail="true" />` remains set to `true`.

```
<system.web>
  <deployment retail="true" />
</system.web>
```

Remediation:

1. Open the `machine.config` file located in:
   `%systemroot%\\Microsoft.NET\\Framework<bitness (if not the 32 bit)>\\<framework version>\\CONFIG`
2. Add the line `<deployment retail="true" />` within the `<system.web>` section
3. If systems are 64-bit, do the same for the `machine.config` located in:
   `%systemroot%\\Microsoft.NET\\Framework<bitness (if not the 32 bit)>\\<framework version>\\CONFIG`

Default Value:

The `<deployment retail>` tag is not included in the `machine.config` by default.

References:

1. http://msdn.microsoft.com/en-US/library/ms228298%28VS.80%29.aspx

CIS Controls:

18.9 Sanitize Deployed Software Of Development Artifacts
For in-house developed applications, ensure that development artifacts (sample data and scripts; unused libraries, components, debug code; or tools) are not included in the deployed software, or accessible in the production environment.

3.2 Ensure 'debug' is turned off (Scored)

Profile Applicability:

- Level 2 - IIS 10

Description:

Developers often enable the debug mode during active ASP.NET development so that they do not have to continually clear their browsers cache every time they make a change to a resource handler. The problem would arise from this being left "on" or set to "true". Compilation debug output is displayed to the end user, allowing malicious persons to obtain detailed information about applications.

This is a defense in depth recommendation due to the `<deployment retail="true" />` in the `machine.config` configuration file overriding any debug settings. It is recommended that debugging still be turned off.

Rationale:

Setting `<compilation debug>` to `false` ensures that detailed error information does not inadvertently display during live application usage, mitigating the risk of application information leakage falling into unscrupulous hands.

Audit:

Browse to and open the `web.config` file pertaining to the server or specific application that has been configured. Locate the `<compilation debug>` switch and verify it is set to false.

```
<configuration>
  <system.web>
    <compilation debug="false" />
  </system.web>
</configuration>
```

Remediation:

To use the UI to make this change:

1. Open IIS Manager and navigate desired server, site, or application
2. In Features View, double-click .NET Compilation
3. On the .NET Compilation page, in the Behavior section, ensure the Debug field is set to False
4. When finished, click Apply in the Actions pane

Note: The `<compilation debug>` switch will not be present in the `web.config` file unless it has been added manually, or has previously been configured using the IIS Manager GUI.

Default Value:

The compilation of debug binaries is not enabled by default.

References:

1. http://technet.microsoft.com/en-us/library/cc725812%28WS.10%29.aspx

CIS Controls:

18.9 Sanitize Deployed Software Of Development Artifacts
For in-house developed applications, ensure that development artifacts (sample data and scripts; unused libraries, components, debug code; or tools) are not included in the deployed software, or accessible in the production environment.

3.3 Ensure custom error messages are not off (Scored)

Profile Applicability:

- Level 2 - IIS 10

Description:

When an ASP.NET application fails and causes an HTTP/1.x 500 Internal Server Error, or a feature configuration (such as Request Filtering) prevents a page from being displayed, an error message will be generated. Administrators can choose whether or not the application should display a friendly message to the client, detailed error message to the client, or detailed error message to localhost only. The `<customErrors>` tag in the `web.config` has three modes:

- On: Specifies that custom errors are enabled. If no `defaultRedirect` attribute is specified, users see a generic error. The custom errors are shown to the remote clients and to the local host
- Off: Specifies that custom errors are disabled. The detailed ASP.NET errors are shown to the remote clients and to the local host
- RemoteOnly: Specifies that custom errors are shown only to the remote clients, and that ASP.NET errors are shown to the local host. This is the default value

This is a defense in depth recommendation due to the `<deployment retail="true" />` in the `machine.config` file overriding any settings for `customErrors` to be turned `Off`. It is recommended that `customErrors` still be turned to `On`or `RemoteOnly`.

Rationale:

`customErrors` can be set to `On`or `RemoteOnly` without leaking detailed application information to the client. Ensuring that `customErrors` is not set to `Off`will help mitigate the risk of malicious persons learning detailed application error and server configuration information.

Audit:

Find and open the `web.config` file for the application/site and verify that the tag has either `<customErrors mode="RemoteOnly" />` or `<customErrors mode="On" />` defined.

Remediation:

`customErrors` may be set for a server, site, or application using the IIS Manager GUI, using `AppCmd.exe` commands in a command-line window, directly editing the configuration files,

or by writing WMI scripts. Perform the following to set the `customErrors` mode to `RemoteOnly` or `On` for a Web Site in the IIS Manager GUI:

1. Open the IIS Manager GUI and navigate to the site to be configured
2. In Features View, find and double-click .NET Error Pages icon
3. In the Actions Pane, click Edit Feature Settings
4. In modal dialog, choose On or Remote Only for Mode settings
5. Click OK

Default Value:

The default value is `<customErrors mode= "RemoteOnly" />`.

References:

1. http://technet.microsoft.com/en-us/library/dd569096%28WS.10%29.aspx

CIS Controls:

18.5 Sanitize Output Of Applications
Do not display system error messages to end-users (output sanitization).

3.4 Ensure IIS HTTP detailed errors are hidden from displaying remotely (Scored)

Profile Applicability:

- Level 1 - IIS 10

Description:

A Web site's error pages are often set to show detailed error information for troubleshooting purposes during testing or initial deployment. To prevent unauthorized users from viewing this privileged information, detailed error pages must not be seen by remote users. This setting can be modified in the `errorMode` attribute setting for a Web site's error pages. By default, the `errorMode` attribute is set in the `Web.config` file for the Web site or application and is located in the `<httpErrors>` element of the `<system.webServer>` section. It is recommended that custom errors be prevented from displaying remotely.

Rationale:

The information contained in custom error messages can provide clues as to how applications function, opening up unnecessary attack vectors. Ensuring custom errors are never displayed remotely can help mitigate the risk of malicious persons obtaining information as to how the application works.

Audit:

The `errorMode` attribute is set in the `Web.config` file for the Web site or application in the `<httpErrors>` element of the `<system.webServer>` section. Browse to the `web.config` and verify the `errorMode` is set to `DetailedLocalOnly` or `Custom`:

```
<system.web>
  <system.webServer>
    <httpErrors errorMode="DetailedLocalOnly">
    </httpErrors>
  </system.webServer>
</system.web>
```

Remediation:

The following describes how to change the `errorMode` attribute to `DetailedLocalOnly` or `Custom` for a Web site by using IIS Manager:

1. Open IIS Manager with Administrative privileges

2. In the Connections pane on the left, expand the server, then expand the Sites folder
3. Select the Web site or application to be configured
4. In Features View, select Error Pages, in the Actions pane, select Open Feature
5. In the Actions pane, select Edit Feature Settings
6. In the Edit Error Pages Settings dialog, under Error Responses, select either Custom error pages or Detailed errors for local requests and custom error pages for remote requests
7. Click OK and exit the Edit Error Pages Settings dialog

Default Value:

The default `errorMode` is DetailedLocalOnly.

References:

1. http://technet.microsoft.com/en-us/library/dd391900%28WS.10%29.aspx
2. http://www.iis.net/configreference/system.webserver/httperrors

CIS Controls:

18.5 Sanitize Output Of Applications
Do not display system error messages to end-users (output sanitization).

3.5 Ensure ASP.NET stack tracing is not enabled (Scored)

Profile Applicability:

- Level 2 - IIS 10

Description:

The `trace` element configures the ASP.NET code tracing service that controls how trace results are gathered, stored, and displayed. When tracing is enabled, each page request generates trace messages that can be appended to the page output or stored in an application trace log.

This is a defense in depth recommendation due to the `<deployment retail="true" />` in the `machine.config` file overriding any settings for ASP.NET stack tracing that are left on. It is recommended that ASP.NET stack tracing still be turned off.

Rationale:

In an active Web Site, tracing should not be enabled because it can display sensitive configuration and detailed stack trace information to anyone who views the pages in the site. If necessary, the `localOnly` attribute can be set to true to have trace information displayed only for localhost requests. Ensuring that ASP.NET stack tracing is not on will help mitigate the risk of malicious persons learning detailed stack trace information.

Audit:

Tracing is configurable at numerous levels:

1. Machine.config
2. Root-level web.config
3. Application-level web.config
4. Virtual or physical directory-level web.config
5. Individual ASP.Net page level

Verify ASP.NET tracing is not turned on, via a per-page basis in the application.

Ensure the trace attribute is not enabled:

```
Trace="true"
```

On an application basis like in the `web.config` ensure that tracing is not enabled like:

```
<configuration>
  <system.web>
```

```
     <trace enabled="true">
  </system.web>
</configuration>
```

Remediation:

1. Ensure `<deployment retail="true" />`is enabled in the `machine.config`.
2. Remove all attribute references to ASP.NET tracing by deleting the trace and trace enable attributes.

Per Page:

Remove any references to:

```
Trace="true"
```

Per Application:

```
<configuration>
  <system.web>
     <trace enabled="true">
  </system.web>
</configuration>
```

Default Value:

The default value for ASP.NET tracing is off.

References:

1. http://msdn.microsoft.com/en-us/library/94c55d08%28v=vs.100%29.aspx
2. http://msdn.microsoft.com/en-us/library/0x5wc973%28v=vs.100%29.aspx

CIS Controls:

18.5 Sanitize Output Of Applications
Do not display system error messages to end-users (output sanitization).

3.6 Ensure 'httpcookie' mode is configured for session state (Scored)

Profile Applicability:

- Level 2 - IIS 10

Description:

A session cookie associates session information with client information for that session, which can be the duration of a user's connection to a site. The cookie is passed in a HTTP header together with all requests between the client and server.

Session information can also be stored in the URL. However, storing session information in this manner has security implications that can open attack vectors such as session hijacking. An effective method used to prevent session hijacking attacks is to force web applications to use cookies to store the session token. This is accomplished by setting the `cookieless` attribute of the `sessionState` node to `UseCookies` or `False` which will in turn keep session state data out of URI. It is recommended that session state be configured to `UseCookies`.

Rationale:

Cookies that have been properly configured help mitigate the risk of attacks such as session hi-jacking attempts by preventing ASP.NET from having to move session information to the URL; moving session information in URI causes session IDs to show up in proxy logs, and is accessible to client scripting via `document.location`.

Audit:

Find and open the `web.config` file for the application/site and verify that the `sessionState` tag is set to use cookies:

```
<system.web>
  <sessionState cookieless="UseCookies" />
</system.web>
```

Remediation:

`SessionState` can be set to `UseCookies` by using the IIS Manager GUI, using `AppCmd.exe` commands in a command-line window, directly editing the configuration files, or by writing WMI scripts. Perform the following to set the `cookieless` attribute of the `sessionState` node to `UseCookies` in the IIS Manager GUI:

1. Open the IIS Manager GUI and navigate desired server, site, or application

2. In Features View, find and double-click the Session State icon
3. In the Cookie Settings section, choose Use Cookies from the Mode dropdown
4. In the Actions Pane, click Apply

To use `AppCmd.exe` to configure `sessionState` at the server level, the command would look like this:

```
%systemroot%\system32\inetsrv\appcmd set config /commit:WEBROOT
/section:sessionState /cookieless:UseCookies /cookieName:ASP.NET_SessionID
/timeout:20
```

When `Appcmd.exe` is used to configure the `<sessionstate>` element at the global level in IIS, the `/commit:WEBROOT` switch must be included so that configuration changes are made to the root `web.config` file instead of `ApplicationHost.config`.

Default Value:

By default, IIS maintains session state data for a managed code application in the worker process where the application runs e.g. In Process.

References:

1. http://www.iis.net/learn/application-frameworks/scenario-build-an-aspnet-website-on-iis/planning-step-2-plan-asp-net-settings
2. http://msdn.microsoft.com/en-us/library/h6bb9cz9%28VS.71%29.aspx

CIS Controls:

18 Application Software Security
Application Software Security

3.7 Ensure 'cookies' are set with HttpOnly attribute (Scored)

Profile Applicability:

- Level 1 - IIS 10

Description:

The `httpOnlyCookies` attribute of the `httpCookies` node determines if IIS will set the `HttpOnly` flag on HTTP cookies it sets. The `HttpOnly` flag indicates to the user agent that the cookie must not be accessible by client-side script (i.e document.cookie). It is recommended that the `httpOnlyCookies` attribute be set to `true`.

Rationale:

When cookies are set with the `HttpOnly` flag, they cannot be accessed by client side scripting running in the user's browser. Preventing client-side scripting from accessing cookie content may reduce the probability of a cross site scripting attack materializing into a successful session hijack.

Audit:

After the next time IIS is restarted, browse to and open the `web.config` for the application in which `httpOnly` cookies have been turned on. Confirm the `httpOnlyCookies`attribute is set to true: `<httpCookies httpOnlyCookies="true" />`.

Remediation:

1. Locate and open the application's `web.config` file
2. Add the `<httpCookies httpOnlyCookies="true" />` tag within `<system.web>`:

```
<configuration>
  <system.web>
    <httpCookies httpOnlyCookies="true" />
  </system.web>
</configuration>
```

Setting the value of the `httpOnlyCookies` attribute of the `httpCookies` element to `true` will add the `HttpOnly` flag to all the cookies set by the application. All modern versions of browsers recognize `HttpOnly` attribute; older versions will either treat them as normal cookies or simply ignore them altogether.

Default Value:

By default, ASP.NET 2.0 does not force cookies to `httpOnly`.

References:

1. https://tools.ietf.org/wg/httpstate/charters
2. https://www.owasp.org/index.php/HTTPOnly#Browsers_Supporting_HttpOnly
3. https://msdn.microsoft.com/en-us/library/ms533046.aspx

CIS Controls:

18 Application Software Security
Application Software Security

3.8 Ensure 'MachineKey validation method - .Net 3.5' is configured (Scored)

Profile Applicability:

- Level 2 - IIS 10

Description:

The `machineKey` element of the ASP.NET `web.config` specifies the algorithm and keys that ASP.NET will use for encryption. The Machine Key feature can be managed to specify hashing and encryption settings for application services such as view state, Forms authentication, membership and roles, and anonymous identification.

The following validation methods are available:

- Advanced Encryption Standard (AES) is relatively easy to implement and requires little memory. AES has a key size of 128, 192, or 256 bits. This method uses the same private key to encrypt and decrypt data, whereas a public-key method must use a pair of keys
- Message Digest 5 (MD5) is used for digital signing of applications. This method produces a 128-bit message digest, which is a compressed form of the original data
- Secure Hash Algorithm (SHA1) is considered more secure than MD5 because it produces a 160-bit message digest
- Triple Data Encryption Standard (TripleDES) is a minor variation of Data Encryption Standard (DES). It is three times slower than regular DES but can be more secure because it has a key size of 192 bits. If performance is not a primary consideration, consider using TripleDES

It is recommended that AES or SHA1 methods be configured for use at the global level.

Rationale:

Setting the validation property to AES will provide confidentiality and integrity protection to the viewstate. AES is the strongest encryption algorithm supported by the validation property. Setting the validation property to SHA1 will provide integrity protection to the viewstate. SHA1 is the strongest hashing algorithm supported by the validation property.

Audit:

To verify the Machine Key validation method using IIS Manager:

1. Open IIS Manager and navigate to the level that was configured, the WEBROOT, or server in this case

2. In the features view, double click Machine Key
3. On the Machine Key page, verify that SHA1 is selected in the validation method dropdown

Remediation:

Machine key encryption can be set by using the UI, running `appcmd.exe` commands, by editing configuration files directly, or by writing WMI scripts. To set the Machine Key encryption at the global level using an `appcmd.exe` command:

```
%systemroot%\system32\inetsrv\appcmd set config /commit:WEBROOT
/section:machineKey /validation:SHA1
```

Note: When `Appcmd.exe` is used to configure the `<machineKey>` element at the global level in IIS, the `/commit:WEBROOT` switch must be included so that configuration changes are made to the root `web.config` file instead of `ApplicationHost.config`.

Default Value:

The default Machine Key validation method is SHA1.

References:

1. http://technet.microsoft.com/en-us/library/cc772271%28WS.10%29.aspx
2. http://technet.microsoft.com/en-us/library/cc772287%28WS.10%29.aspx

CIS Controls:

14.2 Encrypt All Sensitive Information Over Less-trusted Networks
All communication of sensitive information over less-trusted networks should be encrypted. Whenever information flows over a network with a lower trust level, the information should be encrypted.

3.9 Ensure 'MachineKey validation method - .Net 4.5' is configured (Scored)

Profile Applicability:

- Level 1 - IIS 10

Description:

The `machineKey` element of the ASP.NET `web.config` specifies the algorithm and keys that ASP.NET will use for encryption. The Machine Key feature can be managed to specify hashing and encryption settings for application services such as view state, Forms authentication, membership and roles, and anonymous identification.

The following validation methods are available:

- Advanced Encryption Standard (AES) is relatively easy to implement and requires little memory. AES has a key size of 128, 192, or 256 bits. This method uses the same private key to encrypt and decrypt data, whereas a public-key method must use a pair of keys
- Message Digest 5 (MD5) is used for digital signing of applications. This method produces a 128-bit message digest, which is a compressed form of the original data
- Secure Hash Algorithm (SHA1) is considered more secure than MD5 because it produces a 160-bit message digest
- Triple Data Encryption Standard (TripleDES) is a minor variation of Data Encryption Standard (DES). It is three times slower than regular DES but can be more secure because it has a key size of 192 bits. If performance is not a primary consideration, consider using TripleDES
- Secure Hash Algorithm (SHA-2) is a family of two similar hash functions, with different block sizes known as SHA-256 and SHA-512. They differ in the word size; SHAS-256 used 32-bit words and SHA-512 uses 64-bit words.

It is recommended that SHA-2 methods be configured for use at the global level.

Rationale:

SHA-2 is the strongest hashing algorithm supported by the validation property so it should be used as the validation method for the MachineKey in .Net 4.5.

Audit:

To verify the Machine Key validation method using IIS Manager:

1. Open IIS Manager and navigate to the level that was configured, the WEBROOT, or server in this case
2. In the features view, double click Machine Key
3. On the Machine Key page, verify that HMACSHA256 is selected in the validation method dropdown

Remediation:

Machine key encryption can be set by using the UI, running `appcmd.exe` commands, by editing configuration files directly, or by writing WMI scripts. To set the Machine Key encryption at the global level using an `appcmd.exe` command:

```
%systemroot%\system32\inetsrv\appcmd set config /commit:WEBROOT
/section:machineKey /validation:HMACSHA256
```

Note: When `Appcmd.exe` is used to configure the `<machineKey>` element at the global level in IIS, the `/commit:WEBROOT` switch must be included so that configuration changes are made to the root `web.config` file instead of `ApplicationHost.config`.

Default Value:

The default Machine Key validation method is SHA256.

References:

1. http://www.iis.net/learn/get-started/whats-new-in-iis-8/iis-80-aspnet-configuration-management

CIS Controls:

14.2 Encrypt All Sensitive Information Over Less-trusted Networks
All communication of sensitive information over less-trusted networks should be encrypted. Whenever information flows over a network with a lower trust level, the information should be encrypted.

3.10 Ensure global .NET trust level is configured (Scored)

Profile Applicability:

- Level 1 - IIS 10

Description:

This only applies to .Net 2.0. Future versions have stopped supporting this feature.

An application's trust level determines the permissions that are granted by the ASP.NET code access security (CAS) policy. CAS defines two trust categories: full trust and partial trust. An application that has full trust permissions may access all resource types on a server and perform privileged operations, while applications that run with partial trust have varying levels of operating permissions and access to resources.

The possible values for the Level property of the TrustSection class are:

- Full: Specifies unrestricted permissions and grants the ASP.NET application permissions to access any resource that is subject to operating system security; all privileged operations are supported
- High: specifies a high level of code access security which limits the application from doing the following:
    - Call unmanaged code
    - Call serviced components
    - Write to the event log
    - Access Microsoft Windows Message Queuing queues
    - Access ODBC, OLD DB, or Oracle data sources
- Medium: specifies a medium level of code access security, which means that in addition to the restrictions for High, the ASP.NET application cannot do any of the following things:
    - Access files outside the application directory
    - Access the registry
- Low: specifies a low level of code access security, which means that in addition to the restrictions for Medium, the application is prevented from performing any of the following actions:
    - Write to the file system
    - Call the `System.Security.CodeAccessPermission.Assert` method to expand permissions to resources
    - Minimal: specifies a minimal level of code access security, which means that the application has only execute permission

It is recommended that the global .NET Trust Level be set to Medium or lower.

Rationale:

The CAS determines the permissions that are granted to the application on the server. Setting a minimal level of trust that is compatible with the applications will limit the potential harm that a compromised application could cause to a system.

Audit:

To verify the global .NET Trust Level using IIS Manager:

1. Open IIS Manager and navigate to the level that was configured, the server in this example
2. In the features view, double click .NET Trust Levels
3. On the .NET Trust Levels page, verify that `Medium (web\_mediumtrust.config)` is selected in the Trust Level dropdown

Remediation:

Trust level can be set by using the UI, running `appcmd.exe` commands, by editing configuration files directly, or by writing WMI scripts. To set the .Net Trust Level to Medium at the server level using an `appcmd.exe` command:

```
%systemroot%\system32\inetsrv\appcmd set config /commit:WEBROOT
/section:trust /level:Medium
```

When `Appcmd.exe` is used to configure the element at the global level in IIS, the `/commit:WEBROOT` switch must be included so that configuration changes are made to the root `web.config` file instead of `ApplicationHost.config`.

Default Value:

By default, ASP.NET web applications run under the full trust setting.

References:

1. http://technet.microsoft.com/en-us/library/cc772237(WS.10).aspx
2. http://msdn.microsoft.com/en-us/library/ms691448%28VS.90%29.aspx
3. Professional IIS 7 by Ken Schaefer, Jeff Cochran, Scott Forsyth, Rob Baugh, Mike Everest, Dennis Glendenning
4. http://support.microsoft.com/kb/2698981

CIS Controls:

14.4 Protect Information With Access Control Lists
All information stored on systems shall be protected with file system, network share, claims, application, or database specific access control lists. These controls will enforce the

principal that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

4 Request Filtering and Other Restriction Modules

Introduced in IIS 7.0 for the first time, Request Filtering is a powerful module that provides a configurable set of rules that enables administrators to allow or reject the types of requests that they determine should be allowed or rejected at the server, web site, or web application levels.

Earlier versions of Internet Information Services provided the tool UrlScan, which was provided as an add-on to enable system administrators to enforce tighter security policies on their web servers. All of the core features of URLScan have been incorporated into the Request Filtering module. Due to the close nature of functionality in these two tools, reference to legacy URLScan settings will be made where applicable.

IIS 8 also introduced modules for Dynamic IP Address Restrictions. This module can be configured to automatically block web site access based on specific rules.

Note: Request Filtering and IP and Domain Restrictions must be enabled as a role service under IIS in order to configure any of its features.

4.1 Ensure 'maxAllowedContentLength' is configured (Not Scored)

Profile Applicability:

- Level 2 - IIS 10

Description:

The `maxAllowedContentLength` Request Filter is the maximum size of the http request, measured in bytes, which can be sent from a client to the server. Configuring this value enables the total request size to be restricted to a configured value. It is recommended that the overall size of requests be restricted to a maximum value appropriate for the server, site, or application.

Rationale:

Setting an appropriate value that has been tested for the `maxAllowedContentLength` filter will lower the impact an abnormally large request would otherwise have on IIS and/or web applications. This helps to ensure availability of web content and services, and may also help mitigate the risk of buffer overflow type attacks in unmanaged components.

Audit:

Upon exceeding the configured value set for the Request Filter, IIS will throw a Status Code 404.13.

To manually verify the change, locate and open the `web.config` for the web site or application in which the request filter was set. Ensure the value defined for `maxAllowedContentLength` is what was set. The 28.6MB max example would show:

```
<configuration>
  <system.webServer>
    <security>
      <requestFiltering>
        <requestLimits
           maxAllowedContentLength="30000000" />
      </requestFiltering>
    </security>
  </system.webServer>
</configuration>
```

Remediation:

The `MaxAllowedContentLength` Request Filter may be set for a server, website, or application using the IIS Manager GUI, using `AppCmd.exe` commands in a command-line window, and/or directly editing the configuration files. To configure using the IIS Manager GUI:

1. Open Internet Information Services (IIS) Manager
2. In the Connections pane, click on the server, site, application, or directory to be configured
3. In the Home pane, double-click Request Filtering
4. Click Edit Feature Settings... in the Actions pane
5. Under the Request Limits section, key the maximum content length in bytes that will allow applications to retain their intended functionality, such as 30000000 (approx. 28.6 MB)

To set this Request Filter using an `AppCmd.exe` command, run the following command at an elevated command prompt:

```
%systemroot%\system32\inetsrv\appcmd set config /section:requestfiltering
/requestLimits.maxAllowedContentLength:30000000
```

Default Value:

When request filtering is installed on a system, the default value is: `maxAllowedContentLength="30000000"`, which is approximately 28.6MB.

References:

1. http://www.iis.net/ConfigReference/system.webServer/security/requestFiltering/requestLimits
2. http://learn.iis.net/page.aspx/143/use-request-filtering/

CIS Controls:

18 Application Software Security
Application Software Security

4.2 Ensure 'maxURL request filter' is configured (Scored)

Profile Applicability:

- Level 2 - IIS 10

Description:

The `maxURL` attribute of the `<requestLimits>` property is the maximum length (in Bytes) in which a requested URL can be (excluding query string) in order for IIS to accept. Configuring this Request Filter enables administrators to restrict the length of the requests that the server will accept. It is recommended that a limit be put on the length of URL.

Rationale:

With a properly configured Request Filter limiting the amount of data accepted in the URL, chances of undesired application behaviors affecting the availability of content and services are reduced.

Audit:

IIS will log a 404.14 HTTP status if the requested URL was rejected because it exceeded the length defined in the filter.

To manually verify the change, locate and open the `web.config` for the web site or application in which the request filter was set. Verify the value defined for maxURL.

```
<configuration>
  <system.webServer>
    <security>
      <requestFiltering>
        <requestLimits
           maxURL="4096" />
      </requestFiltering>
    </security>
  </system.webServer>
</configuration>
```

Remediation:

The `MaxURL` Request Filter may be set for a server, website, or application using the IIS Manager GUI, using `AppCmd.exe` commands in a command-line window, and/or directly editing the configuration files. To configure using the IIS Manager GUI:

1. Open Internet Information Services (IIS) Manager

2. In the Connections pane, click on the connection, site, application, or directory to be configured
3. In the Home pane, double-click Request Filtering
4. Click Edit Feature Settings... in the Actions pane
5. Under the Request Limits section, key the maximum URL length in bytes that has been tested with web applications

To set this Request Filter using an `AppCmd.exe` command, run the following command at an elevated command prompt:

```
%systemroot%\system32\inetsrv\appcmd set config /section:requestfiltering
/requestLimits.maxURL:4096
```

Default Value:

When Request Filtering is installed on a system, the default value for `maxURL="4096"`.

References:

1. http://www.iis.net/ConfigReference/system.webServer/security/requestFiltering/requestLimits
2. http://learn.iis.net/page.aspx/143/use-request-filtering/

CIS Controls:

18 Application Software Security
Application Software Security

4.3 Ensure 'MaxQueryString request filter' is configured (Scored)

Profile Applicability:

- Level 2 - IIS 10

Description:

The `MaxQueryString` Request Filter describes the upper limit on the length of the query string that the configured IIS server will allow for websites or applications. It is recommended that values always be established to limit the amount of data will can be accepted in the query string.

Rationale:

With a properly configured Request Filter limiting the amount of data accepted in the query string, chances of undesired application behaviors such as app pool failures are reduced.

Audit:

If a request is rejected because it exceeds the value set in the `maxQueryString` Request Filter, a 404.15 HTTP status is logged to the IIS log file.

To manually verify the change, locate and open `the web.config` for the web site or application in which the filter was set. Ensure the value defined for `maxQueryString` is what was configured.

```
<configuration>
  <system.webServer>
    <security>
      <requestFiltering>
        <requestLimits
           maxQueryString="2048" />
      </requestFiltering>
    </security>
  </system.webServer>
</configuration>
```

Remediation:

The `MaxQueryString` Request Filter may be set for a server, website, or application using the IIS Manager GUI, using `AppCmd.exe` commands in a command-line window, and/or directly editing the configuration files. To configure using the IIS Manager GUI:

1. Open Internet Information Services (IIS) Manager

2. In the Connections pane, go to the connection, site, application, or directory to be configured
3. In the Home pane, double-click Request Filtering
4. Click Edit Feature Settings... in the Actions pane
5. Under the Request Limits section, key in a safe upper bound in the Maximum query string (Bytes) textbox

To set this Request Filter using an `AppCmd.exe` command, run the following command at an elevated command prompt:

```
%systemroot%\system32\inetsrv\appcmd set config /section:requestfiltering
/requestLimits.maxQueryString:2048
```

Default Value:

When request filtering is installed on a system, the default value is `maxQueryString="2048"`.

References:

1. http://www.iis.net/ConfigReference/system.webServer/security/requestFiltering/requestLimits
2. http://learn.iis.net/page.aspx/143/use-request-filtering/

CIS Controls:

18 Application Software Security
Application Software Security

4.4 Ensure non-ASCII characters in URLs are not allowed (Scored)

Profile Applicability:

- Level 2 - IIS 10

Description:

This feature is used to allow or reject all requests to IIS that contain non-ASCII characters. When using this feature, Request Filtering will deny the request if high-bit characters are present in the URL. The UrlScan equivalent is `AllowHighBitCharacters`. It is recommended that requests containing non-ASCII characters be rejected, where possible.

Rationale:

This feature can help defend against canonicalization attacks, reducing the potential attack surface of servers, sites, and/or applications.

Audit:

If a request is rejected because it contains a high-bit character, a 404.12 HTTP status is logged to the IIS log file.

To manually verify the change, locate and open the `web.config` for the web site or application in which the request filter was set. Ensure the value defined for the filter is false, as such:

```
<configuration>
  <system.webServer>
    <security>
      <requestFiltering
         allowHighBitCharacters="false">
      </requestFiltering>
    </security>
  </system.webServer>
</configuration>
```

Remediation:

The `AllowHighBitCharacters` Request Filter may be set for a server, website, or application using the IIS Manager GUI, using `AppCmd.exe` commands in a command-line window, and/or directly editing the configuration files. To configure using the IIS Manager GUI:

1. Open Internet Information Services (IIS) Manager

2. In the Connections pane, go to the connection, site, application, or directory to be configured
3. In the Home pane, double-click Request Filtering
4. Click Edit Feature Settings... in the Actions pane
5. Under the General section, uncheck Allow high-bit characters

Note: Disallowing high-bit ASCII characters in the URL may negatively impact the functionality of sites requiring international language support.

To set this Request Filter using an `AppCmd.exe` command, run the following command at an elevated command prompt:

```
%systemroot%\system32\inetsrv\appcmd set config /section:requestfiltering
/allowHighBitCharacters:false
```

Default Value:

When Request Filtering is installed on a system, the default behavior is to allow high-bit characters in URI.

References:

1. http://learn.iis.net/page.aspx/143/use-request-filtering/
2. http://learn.iis.net/page.aspx/936/urlscan-1-reference/
3. Professional IIS 7 by Ken Schaefer, Jeff Cochran, Scott Forsyth, Rob Baugh, Mike Everest, Dennis Glendenning

CIS Controls:

18 Application Software Security
Application Software Security

4.5 Ensure Double-Encoded requests will be rejected (Scored)

Profile Applicability:

- Level 1 - IIS 10

Description:

This Request Filter feature prevents attacks that rely on double-encoded requests and applies if an attacker submits a double-encoded request to IIS. When the double-encoded requests filter is enabled, IIS will go through a two iteration process of normalizing the request. If the first normalization differs from the second, the request is rejected and the error code is logged as a 404.11. The double-encoded requests filter was the `VerifyNormalization` option in UrlScan. It is recommended that double-encoded requests be rejected.

Rationale:

This feature will help prevent attacks that rely on URLs that have been crafted to contain double-encoded request(s).

Audit:

If a request is rejected because it contains a double-encoded request, a 404.11 HTTP status is logged to the IIS log file.

To manually verify the change, locate and open the `web.config` for the web site or application in which the request filter was set. Ensure the value defined for `allowDoubleEscaping` is `false`:

```
<configuration>
  <system.webServer>
    <security>
      <requestFiltering
         allowDoubleEscaping="false">
      </requestFiltering>
    </security>
  </system.webServer>
</configuration>
```

Remediation:

The `allowDoubleEscaping` Request Filter may be set for a server, website, or application using the IIS Manager GUI, using `AppCmd.exe` commands in a command-line window, and/or directly editing the configuration files. To configure using the IIS Manager GUI:

1. Open Internet Information Services (IIS) Manager
2. In the Connections pane, select the site, application, or directory to be configured
3. In the Home pane, double-click Request Filtering
4. Click Edit Feature Settings... in the Actions pane
5. Under the General section, uncheck Allow double escaping

If a file name in a URL includes "+" then `allowDoubleEscaping` must be set to `true` to allow functionality.

To set this Request Filter using an `AppCmd.exe` command, run the following command at an elevated command prompt:

```
%systemroot%\system32\inetsrv\appcmd set config /section:requestfiltering
/allowDoubleEscaping:false
```

Default Value:

When Request Filtering is installed on a system, the default behavior is to not allow double-encoded requests.

References:

1. http://www.iis.net/ConfigReference/system.webServer/security/requestFiltering/requestLimits
2. http://learn.iis.net/page.aspx/143/use-request-filtering/

CIS Controls:

18 Application Software Security
Application Software Security

4.6 Ensure 'HTTP Trace Method' is disabled (Scored)

Profile Applicability:

- Level 1 - IIS 10

Description:

The HTTP TRACE method returns the contents of client HTTP requests in the entity-body of the TRACE response. Attackers could leverage this behavior to access sensitive information, such as authentication data or cookies, contained in the HTTP headers of the request. One such way to mitigate this is by using the `<verbs>` element of the `<requestFiltering>` collection. The `<verbs>` element replaces the [AllowVerbs] and [DenyVerbs] features in UrlScan. It is recommended the HTTP TRACE method be denied.

Rationale:

Attackers may abuse HTTP TRACE functionality to gain access to information in HTTP headers such as cookies and authentication data. This risk can be mitigated by not allowing the TRACE verb.

Audit:

IIS will return an HTTP 404.6 error to the client when Request Filtering blocks an HTTP request because of a denied HTTP verb. To manually verify the change, browse to the `web.config` file for which the change was made and verify the below configuration:

```
<configuration>
  <system.webServer>
    <security>
       <requestFiltering>
          <verbs>
            <add verb="TRACE" allowed="false" />
          </verbs>
        </requestFiltering>
     </security>
  </system.webServer>
</configuration>
```

Remediation:

1. Open Internet Information Services (IIS) Manager
2. In the Connections pane, select the site, application, or directory to be configured
3. In the Home pane, double-click Request Filtering
4. In the Request Filtering pane, click the HTTP verbs tab, and then click Deny Verb... in the Actions pane

5. In the Deny Verb dialog box, enter the TRACE, and then click OK

To set this Request Filter using an `AppCmd.exe` command, run the following command at an elevated command prompt:

```
%systemroot%\system32\inetsrv\appcmd set config /section:requestfiltering
/+verbs.[verb='TRACE',allowed='false']
```

Default Value:

The TRACE verb is not filtered by default.

References:

1. http://www.kb.cert.org/vuls/id/867593
2. http://www.iis.net/ConfigReference/system.webServer/security/requestFiltering/verbs

CIS Controls:

18 Application Software Security
Application Software Security

4.7 Ensure Unlisted File Extensions are not allowed (Scored)

Profile Applicability:

- Level 1 - IIS 10

Description:

The `FileExtensions` Request Filter allows administrators to define specific extensions their web server(s) will allow and disallow. The property `allowUnlisted` will cover all other file extensions not explicitly allowed or denied. Often times, extensions such as `.config`, `.bat`, `.exe`, to name a few, should never be served. The `AllowExtensions` and `DenyExtensions` options are the UrlScan equivalents. It is recommended that all extensions be unallowed at the most global level possible, with only those necessary being allowed.

Rationale:

Disallowing all but the necessary file extensions can greatly reduce the attack surface of applications and servers.

Audit:

When IIS rejects a request based on a file extensions filter, the error code logged is 404.7.

To manually verify the change, locate and open the `web.config` for the web site or application in which the Request Filter was set. Ensure `<fileExtensions allowUnlisted="false">`. The following `web.config` will disallow any requests for files that do not have `.asp`, `.aspx`, or `.html` as their extension:

```
<configuration>
 <system.webServer>
  <security>
   <requestFiltering>
    <fileExtensions allowUnlisted="false">
     <add fileExtension=".asp" allowed="true" />
     <add fileExtension=".aspx" allowed="true" />
     <add fileExtension=".html" allowed="true" />
    </fileExtensions>
   </requestFiltering>
  </security>
 </system.webServer>
</configuration>
```

Remediation:

The `allowUnlisted` Request Filter may be set for a server, website, or application using the IIS Manager GUI, using `AppCmd.exe` commands in a command-line window, and/or directly editing the configuration files. To configure at the server level using the IIS Manager GUI:

1. Open Internet Information Services (IIS) Manager
2. In the Connections pane, select the server
3. In the Home pane, double-click Request Filtering
4. Click Edit Feature Settings... in the Actions pane
5. Under the General section, uncheck Allow unlisted file name extensions

To set this Request Filter using an `AppCmd.exe` command, run the following command at an elevated command prompt:

```
%systemroot%\system32\inetsrv\appcmd set config /section:requestfiltering
/fileExtensions.allowunlisted:false
```

Default Value:

The default Request Filtering configuration allows all unlisted file extensions to be requested.

References:

1. http://www.iis.net/ConfigReference/system.webServer/security/requestFiltering/requestLimits
2. http://www.iis.net/learn/manage/configuring-security/configure-request-filtering-in-iis

CIS Controls:

18 Application Software Security
Application Software Security

4.8 Ensure Handler is not granted Write and Script/Execute (Scored)

Profile Applicability:

- Level 1 - IIS 10

Description:

Handler mappings can be configured to give permissions to `Read`, `Write`, `Script`, or `Execute` depending on what the use is for - reading static content, uploading files, executing scripts, etc. It is recommended to grant a handler either `Execute/``Script` or `Write` permissions, but not both.

Rationale:

By allowing both `Execute`/`_Script_` and `Write` permissions, a handler can run malicious code on the target server. Ensuring these two permissions are never together will help lower the risk of malicious code being executed on the server.

Audit:

Open the `ApplicationHost.config` file in `%systemroot%\\system32\\inetsrv\\config`. Find the `<handlers>` section and verify that the `accessPolicy` attribute does not contain `Write` when `Script` or `Execute` are present. The following is an acceptable example:

```
<system.webserver>
  <handlers accessPolicy="Read, Script">
  </handlers>
</system.webserver>
```

Remediation:

The `accessPolicy` attribute in the `<handlers>` section of either the `ApplicationHost.config` (server-wide) or `web.config` (site or application) must not have `Write` present when `Script` or `Execute` are present. To resolve this issue for a Web server, the attribute in the `<handlers>` section of the `ApplicationHost.config` file for the server must manually be edited. To edit the ApplicationHost.config file by using Notepad, perform the following steps:

1. Open Notepad as Administrator
2. Open the ApplicationHost.config file in `%systemroot%\\system32\\inetsrv\\config`
3. Edit the `<handlers>` section `accessPolicy` attribute so that `Write` is not present when `Script` or `Execute` are present

To set this Request Filter using an `AppCmd.exe` command, run the following command at an elevated command prompt:

```
%systemroot%\system32\inetsrv\appcmd set config /section:handlers
/accessPolicy:Read,Script
```

Note: This configuration change cannot be made by using IIS Manager.

Default Value:

The default handlers `accessPolicy` is `Read`, `Script`.

References:

1. http://technet.microsoft.com/en-us/library/dd391910%28WS.10%29.aspx
2. http://blogs.iis.net/thomad/archive/2006/11/05/quo-vadis-accessflags.aspx

CIS Controls:

18 Application Software Security
Application Software Security

4.9 Ensure 'notListedIsapisAllowed' is set to false (Scored)

Profile Applicability:

- Level 1 - IIS 10

Description:

The `notListedIsapisAllowed` attribute is a server-level setting that is located in the `ApplicationHost.config` file in the `<isapiCgiRestriction>` element of the `<system.webServer>` section under `<security>`. This element ensures that malicious users cannot copy unauthorized ISAPI binaries to the Web server and then run them. It is recommended that `notListedIsapisAllowed` be set to `false`.

Rationale:

Restricting this attribute to `false`will help prevent potentially malicious ISAPI extensions from being run.

Audit:

Open the `applicationHost.config` file in `%systemroot%\system32\inetsrv\config`. Verify that the `notListedIsapisAllowed` attribute in the `<isapiCgiRestriction>` element is set to `false`:

```
<system.webServer>
  <security>
    <isapiCgiRestriction notListedIsapisAllowed="false">
    </isapiCgiRestriction>
  </security>
</system.webServer>
```

Remediation:

To use IIS Manager to set the `notListedIsapisAllowed` attribute to `false`:

1. Open IIS Manager as Administrator
2. In the Connections pane on the left, select server to be configured
3. In Features View, select ISAPI and CGI Restrictions; in the Actions pane, select Open Feature
4. In the Actions pane, select Edit Feature Settings
5. In the Edit ISAPI and CGI Restrictions Settings dialog, clear the Allow unspecified ISAPI modules check box, if checked
6. Click OK

To set this Request Filter using an `AppCmd.exe` command, run the following command at an elevated command prompt:

```
%systemroot%\system32\inetsrv\appcmd.exe set config  -
section:system.webServer/security/isapiCgiRestriction
/notListedIsapisAllowed:false
```

Default Value:

The default value for `notListedIsapisAllowed` is `false`.

References:

1. http://technet.microsoft.com/en-us/library/dd378846%28WS.10%29.aspx
2. http://www.iis.net/ConfigReference/system.webServer/security/isapiCgiRestriction

CIS Controls:

18 Application Software Security
Application Software Security

4.10 Ensure 'notListedCgisAllowed' is set to false (Scored)

Profile Applicability:

- Level 1 - IIS 10

Description:

The `notListedCgisAllowed` attribute is a server-level setting that is located in the `ApplicationHost.config` file in the `<isapiCgiRestriction>` element of the `<system.webServer>` section under `<security>`. This element ensures that malicious users cannot copy unauthorized CGI binaries to the Web server and then run them. It is recommended that `notListedCgisAllowed` be set to `false`.

Rationale:

Restricting this attribute to `false` will help prevent unlisted CGI extensions, including potentially malicious CGI scripts from being run.

Audit:

Browse to and open the `applicationHost.config` file and verify that the `notListedCgisAllowed` attribute in the `<isapiCgiRestriction>` element is set to `false`:

```
<system.webServer>
  <security>
    <isapiCgiRestriction notListedCgisAllowed="false">
    </isapiCgiRestriction>
  </security>
</system.webServer>
```

Remediation:

To set the `notListedCgisAllowed` attribute to false using IIS Manager:

1. Open IIS Manager as Administrator
2. In the Connections pane on the left, select the server to configure
3. In Features View, select ISAPI and CGI Restrictions; in the Actions pane, select Open Feature
4. In the Actions pane, select Edit Feature Settings
5. In the Edit ISAPI and CGI Restrictions Settings dialog, clear the Allow unspecified CGI modules check box
6. Click OK

To set this Request Filter using an `AppCmd.exe` command, run the following command at an elevated command prompt:

```
%systemroot%\system32\inetsrv\appcmd.exe set config  -
section:system.webServer/security/isapiCgiRestriction
/notListedCgisAllowed:false
```

Default Value:

The default value for `notListedCgisAllowed` is `false`.

References:

1. http://technet.microsoft.com/en-us/library/dd391919%28WS.10%29.aspx

CIS Controls:

18 Application Software Security
Application Software Security

4.11 Ensure 'Dynamic IP Address Restrictions' is enabled (Not Scored)

Profile Applicability:

- Level 1 - IIS 10

Description:

IIS Dynamic IP Address Restrictions capability can be used to thwart DDos attacks. This is complimentary to the IP Addresses and Domain names Restrictions lists that can be manually maintained within IIS. In contrast, Dynamic IP address filtering allows administrators to configure the server to block access for IPs that exceed the specified request threshold. The default action Deny action for restrictions is to return a Forbidden response to the client.

Rationale:

Dynamic IP address filtering allows administrators to configure the server to block access for IPs that exceed the specified number of requests or requests frequency. Ensure that you receive the Forbidden page once the block has been enforced.

Audit:

Access the web server enough times to trigger the IP restriction based on the settings entered.

Remediation:

1. Open IIS Manager.
2. Open the IP Address and Domain Restrictions feature.
3. Click Edit Dynamic Restrictions Settings..
4. Check the Deny IP Address based on the number of concurrent requests and the Deny IP Address based on the number of requests over a period of time boxes. The values can be tweaked as needed for your specific environment.

Default Value:

By default Dynamic IP Restrictions are not enabled.

References:

1. http://www.iis.net/learn/get-started/whats-new-in-iis-8/iis-80-dynamic-ip-address-restrictions

CIS Controls:

9.6 Deploy And Configure Application Firewalls

Place application firewalls in front of any critical servers to verify and validate the traffic going to the server. Any unauthorized services or traffic should be blocked and an alert generated.

5 IIS Logging Recommendations

This section contains recommendations regarding IIS logging that have not been covered in the Basic Configurations section.

5.1 Ensure Default IIS web log location is moved (Scored)

Profile Applicability:

- Level 1 - IIS 10

Description:

IIS will log relatively detailed information on every request. These logs are usually the first item looked at in a security response, and can be the most valuable. Malicious users are aware of this, and will often try to remove evidence of their activities. It is therefore recommended that the default location for IIS log files be changed to a restricted, non-system drive.

Rationale:

Moving IIS logging to a restricted, non-system drive will help mitigate the risk of logs being maliciously altered, removed, or lost in the event of system drive failure(s).

Audit:

To verify web logs are being logged to the new location, open Windows Explorer and browse to the path that was defined. Depending on how the logging was configured, there will be either:

1. A folder containing .log files or
2. .log files in the root of the specified directory

Remediation:

Moving the default log location can be easily accomplished using the Logging feature in the IIS Management UI or `AppCmd.exe`. To change to D:\LogFiles using `AppCmd.exe`:

```
%systemroot%\system32\inetsrv\appcmd set config -section:sites -
siteDefaults.logfile.directory:"D:\LogFiles"
```

Moving log file stores to a non-system drive or partition separate from where web applications run and/or content is served is preferred. Additionally, folder-level NTFS permissions should be set as restrictive as possible; Administrators and SYSTEM are typically the only principals requiring access.

While standard IIS logs can be moved and edited using IIS Manager, additional management tool add-ons are required in order to manage logs generated by other IIS features, such as Request Filtering and IIS Advanced Logging. These add-ons can be obtained using the Web Platform Installer or from Microsoft's site. The HTTPErr logging location can be changed by adding a registry key.

Default Value:

The default location for web logs in IIS is: `%SystemDrive%\inetpub\logs\LogFiles`.

References:

1. https://technet.microsoft.com/en-us/library/cc770709(v=ws.10).aspx?

CIS Controls:

6.3 Ensure Audit Logging Systems Are Not Subject To Loss (i.e. rotation/archive)
Ensure that all systems that store logs have adequate storage space for the logs generated on a regular basis, so that log files will not fill up between log rotation intervals. The logs must be archived and digitally signed on a periodic basis.

5.2 Ensure Advanced IIS logging is enabled (Scored)

Profile Applicability:

- Level 1 - IIS 10

Description:

IIS Advanced Logging is a module which provides flexibility in logging requests and client data. It provides controls that allow businesses to specify what fields are important, easily add additional fields, and provide policies pertaining to log file rollover and Request Filtering. HTTP request/response headers, server variables, and client-side fields can be easily logged with minor configuration in the IIS management console. It is recommended that Advanced Logging be enabled, and the fields which could be of value to the type of business or application in the event of a security incident, be identified and logged.

Rationale:

Many of the fields available in Advanced Logging many can provide extensive, real-time data and details not otherwise obtainable. Developers and security professionals can use this information to identify and remediate application vulnerabilities/attack patterns.

Audit:

Browse to the location of the Advanced Logs and verify .log files are being generated. Note that logs will be written to disk after a non-determined period of time. They can be written into their specified directory immediately if, in the Log Definition, the Publish real-time events and Write to disk options are selected.

Remediation:

IIS Advanced Logging can be configured for servers, Web sites, and directories in IIS Manager. To enable Advanced Logging using the UI:

1. Open Internet Information Services (IIS) Manager
2. Click the server in the Connections pane
3. Double-click the Logging icon on the Home page
4. Click Select Fields

The fields that will be logged need to be configured using the Add or Edit Fields button.

Note: There may be performance considerations depending on the extent of the configuration. Advanced logging requires installation using Web Platform Installer or manually form the download link in the References section.

Default Value:

IIS Advanced Logging is not enabled by default.

References:

1. https://www.iis.net/learn/get-started/whats-new-in-iis-85/enhanced-logging-for-iis85

CIS Controls:

6.2 Ensure Audit Log Settings Support Appropriate Log Entry Formatting
Validate audit log settings for each hardware device and the software installed on it, ensuring that logs include a date, timestamp, source addresses, destination addresses, and various other useful elements of each packet and/or transaction. Systems should record logs in a standardized format such as syslog entries or those outlined by the Common Event Expression initiative. If systems cannot generate logs in a standardized format, log normalization tools can be deployed to convert logs into such a format.

5.3 Ensure 'ETW Logging' is enabled (Not Scored)

Profile Applicability:

- Level 1 - IIS 10

Description:

IIS introduces a new logging method. Administrators can now send logging information to Event Tracing for Windows (ETW)

Rationale:

IIS flushes log information to disk, therefore prior to IIS, administrators do not have access to real-time logging information. Text-based log files can also be difficult and time consuming to process. By enabling ETW, administrators have access to use standard query tools for viewing real-time logging information.

Audit:

Using Message Analyzer, configure the query for Microsoft-Windows-IIS-Logging. Verify you see live logging data by accessing the website.

Remediation:

To configure ETW logging:

1. Open IIS Manager
2. Select the server or site to enable ETW
3. Select Logging.
4. Ensure Log file format is W3C.
5. Select Both log file and ETW event
6. Save your settings.

References:

1. http://www.iis.net/learn/get-started/whats-new-in-iis-85/logging-to-etw-in-iis-85
2. http://blogs.technet.com/b/erezs_iis_blog/archive/2013/07/15/hook-me-up.aspx
3. https://blogs.msdn.microsoft.com/dcook/2015/09/30/etw-overview/
4. https://social.msdn.microsoft.com/Forums/en-US/a1aa1350-41a0-4490-9ae3-9b4520aeb9d4/faq-common-questions-for-etw-and-windows-event-log?forum=etw

CIS Controls:

6.6 Deploy A SIEM OR Log Analysis Tools For Aggregation And Correlation/Analysis
Deploy a SIEM (Security Information and Event Management) or log analytic tools for log aggregation and consolidation from multiple machines and for log correlation and analysis. Using the SIEM tool, system administrators and security personnel should devise profiles of common events from given systems so that they can tune detection to focus on unusual activity, avoid false positives, more rapidly identify anomalies, and prevent overwhelming analysts with insignificant alerts.

6 FTP Requests

This section contains a crucial configuration setting for running file transfer protocol (FTP).

6.1 Ensure FTP requests are encrypted (Scored)

Profile Applicability:

- Level 1 - IIS 10

Description:

The new FTP Publishing Service for IIS supports adding an SSL certificate to an FTP site. Using an SSL certificate with an FTP site is also known as FTP-S or FTP over Secure Socket Layers (SSL). FTP-S is an RFC standard (RFC 4217) where an SSL certificate is added to an FTP site and thereby making it possible to perform secure file transfers.

Rationale:

By using SSL, the FTP transmission is encrypted and secured from point to point and all FTP traffic as well as credentials are thereby guarded against interception.

Audit:

The FTP site will now require the use of FTP-S; test this by attempting to use an FTP client which either does not support FTP-S or is not configured to use FTP-S. If setup was successful, the request will fail. Conversely, open a command prompt from the server and type `ftp localhost`. After entering credentials, the server should return an Access is Denied message.

Remediation:

To secure an existing FTP site using a SSL Certificate, a certificate must first be installed on the system. Production systems should always use a third party certificate from a trusted root, such as VeriSign. Once that certificate is installed for use in IIS, follow the steps below to configure the FTP site for SSL:

1. Open IIS Manager, select the FTP server and choose FTP SSL Settings in the Features View pane
2. Under the SSL Certificate dropdown, choose the X.509 certificate to be configured for use
3. In the SSL Policy section, click the radio button next to Require SSL connections; it is important to require SSL, because allow SSL still permits non-SSL FTP
4. Click Apply in the Actions pane

Default Value:

By default, FTP sites are not SSL enabled.

References:

1. http://www.windowsnetworking.com/articles_tutorials/IIS-FTP-Publishing-Service-Part3.html
2. http://learn.iis.net/page.aspx/304/using-ftp-over-ssl/#03
3. https://tools.ietf.org/html/rfc4217

CIS Controls:

14.2 Encrypt All Sensitive Information Over Less-trusted Networks
All communication of sensitive information over less-trusted networks should be encrypted. Whenever information flows over a network with a lower trust level, the information should be encrypted.

6.2 Ensure FTP Logon attempt restrictions is enabled (Not Scored)

Profile Applicability:

- Level 1 - IIS 10

Description:

IIS introduced a built-in network security feature to automatically block brute force FTP attacks. This can be used to mitigate a malicious client from attempting a brute-force attack on a discovered account, such as the local administrator account.

Rationale:

Successful brute force FTP attacks can allow an otherwise unauthorized user to make changes to data that should not be made. This could allow the unauthorized user to modify website code by uploading malicious software or even changing functionality for items such as online payments.

Audit:

Access your FTP server using the administrator account and an invalid password. Verify after the maximum number of login attempts has been met that you receive a message 'Connection closed by remote host' when trying to access FTP.

Remediation:

1. Open IIS Manager
2. At the server level, open the FTP Logon Attempt Restrictions feature.
3. Check Enable FTP Logon Attempt Restrictions and enter the maximum number of failed attempts and the time period. Enable Deny IP addresses based on the number of failed login attempts.
4. Click Apply

Default Value:

By default, this feature is not enabled when FTP is installed.

References:

1. http://www.iis.net/learn/get-started/whats-new-in-iis-8/iis-80-ftp-logon-attempt-restrictions

CIS Controls:

16.7 Configure Account Lockouts
Use and configure account lockouts such that after a set number of failed login attempts the account is locked for a standard period of time.

7 Transport Encryption

This section contains recommendations for configuring IIS protocols and cipher suites.

For security protocols (SSL, TLS), there are 2 registry paths that control a protocol state in the O/S: TLS client and TLS server. A web server normally acts as the TLS server in that it is serving web content to clients. There are some instances where a web server is configured as a 'client'. An example of a server acting as a client can be seen when there is dynamic content generation. The webserver queries a database to return content specific to a user's request. In this configuration, the web server is acting as a TLS client. In cases such as these, the configured TLS server protocol and cipher suite preferences take precedence over the client's. This behavior is why for the IIS benchmark we require specific protocol settings for a TLS server and only recommend settings for TLS clients.

If SSLv3 registry keys are not set, the O/S defaults take precedence.

For example, to disable SSLv3 protocol on the TLS server, you need to set the following registry key to 0:
`HKLM\System\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\SSL 3.0\Server\Enabled`

To prevent a client from issuing the Hello command over that legacy protocol the following registry must be set to 0:
`HKLM\System\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\SSL 3.0\Client\Enabled`

The fact that the key is named Enabled can be confusing. The setting of the value to either 0 or 1 actually sets the state of the protocol. 0 being disabled and 1 being enabled.

Here are some specifics into how "Enabled" and "DisabledByDefault" registry settings work. The following article, How to restrict the use of certain cryptographic algorithms and protocols in Schannel.dll, provides additional information related to controlling these protocols and ciphers.

Using the "Enabled = 0" registry setting disables the protocol in a way that can't be overridden by application settings. This is the only robust way to prevent the protocol from being used and no additional settings are required. At the same time, using the "DisabledByDefault" registry setting only prevents that protocol from issuing the Hello command over that protocol when an SSL connection with a server is initiated. This O/S level setting can be overwritten by an application which has application specific TLS coding. An example of this can be shown by setting the protocol within a line of code in your .Net 4.5 application: ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls12. This can override the O/S setting if the DisabledByDefault key

is present. “DisabledByDefault” is useful in the case when you want to have some control over the system settings but also allow an application to explicitly specify the protocols they would like to use.

Enabled only works strongly in the negative case (“Enabled = 0”). If “Enabled=1” or is not set, then “DisabledByDefault” will override in the case where the application takes the system defaults. “Enabled=1” is also overridden by application specific protocol flags.

7.1 Ensure HSTS Header is set (Not Scored)

Profile Applicability:

- Level 2 - IIS 10

Description:

HTTP Strict Transport Security (HSTS) allows a site to inform the user agent to communicate with the site only over HTTPS. This header takes two parameters: max-age, "specifies the number of seconds, after the reception of the STS header field, during which the user agent regards the host (from whom the message was received) as a Known HSTS Host [speaks only HTTPS]"; and includeSubDomains. includeSubDomains is an optional directive that defines how this policy is applied to subdomains. If includeSubDomains is included in the header, it provides the following definition: this HSTS Policy also applies to any hosts whose domain names are subdomains of the Known HSTS Host's domain name.

Rationale:

HTTP Strict Transport Security (HSTS) is a simple and widely supported standard to protect visitors by ensuring that their browsers always connect to a website over HTTPS. HSTS exists to remove the need for the common, insecure practice of redirecting users from http:// to https:// URLs. HSTS relies on the User Agent/Browser to enforce the required behavior. All major browsers support it. If the browser doesn't support HSTS, it will be ignored.

When a browser knows that a domain has enabled HSTS, it does two things:

1. Always uses an https:// connection, even when clicking on an http:// link or after typing a domain into the location bar without specifying a protocol.
2. Removes the ability for users to click through warnings about invalid certificates.

A domain instructs browsers that it has enabled HSTS by returning an HTTP header over an HTTPS connection.

Audit:

The recommended max age is 8 minutes (480 seconds) or greater. Any value greater than 0 is acceptable. Perform the following in IIS Manager to view host headers configured for the server:

1. Open IIS Manager
2. In the Connections pane, select your server
3. In the Features View pane, double click HTTP Response Headers
4. Verify an entry exists named Strict-Transport-Security
5. Double click Strict-Transport-Security and verify the Value: box contains any value greater than 0
6. Click OK.

Perform the following in IIS Manager to view host headers configured for the Website:

1. Open IIS Manager
2. In the Connections pane, expand the tree and select Website
3. In the Features View pane, double click HTTP Response Headers
4. Verify an entry exists name Strict-Transport-Security
5. Double click Strict-Transport-Security and verify the Value: box contains any value greater than 0
6. Click OK.

Remediation:

Any value greater than 0 meets this recommendation. The examples below are specific to 8 minutes but can be adjusted to meet your requirements.

To set the HTTP Header at the server level using an `AppCmd.exe` command, run the following command from an elevated command prompt:

```
%systemroot%\system32\inetsrv\appcmd.exe set config -
section:system.webServer/httpProtocol /+"customHeaders.[name='Strict-
Transport-Security',value='max-age=480']"
```

To set the HTTP Header and include subdomains at the server level using an `AppCmd.exe` command, run the following command from an elevated command prompt:

```
%systemroot%\system32\inetsrv\appcmd.exe set config -
section:system.webServer/httpProtocol /+"customHeaders.[name='Strict-
Transport-Security',value='max-age=480; includeSubDomains']"
```

To set the HTTP Header at the Website level using an `AppCmd.exe` command, run the following command from an elevated command prompt:

```
%systemroot%\system32\inetsrv\appcmd.exe set config -
section:system.webServer/httpProtocol /+"customHeaders.[name='Strict-
Transport-Security',value='max-age=480']"
```

To set the HTTP Header and include subdomains at the Website level using an `AppCmd.exe` command, run the following command from an elevated command prompt:

```
%systemroot%\system32\inetsrv\appcmd.exe set config "<em>Website"</em> -
section:system.webServer/httpProtocol /+"customHeaders.[name='Strict-
Transport-Security',value='max-age=480; includeSubDomains']"
```

References:

1. http://tools.ietf.org/html/rfc6797#section-5.1
2. https://https.cio.gov/hsts/
3. https://www.iis.net/configreference/system.webserver/httpprotocol/customheaders#006

CIS Controls:

18 Application Software Security
Application Software Security

7.2 Ensure SSLv2 is disabled (Scored)

Profile Applicability:

- Level 1 - IIS 10

Description:

This protocol is not considered cryptographically secure. Disabling it is recommended. This protocol is disabled by default if the registry key is not present. A reboot is required for these changes to be reflected.

Rationale:

Disabling weak protocols will help ensure the confidentiality and integrity of in-transit data.

Audit:

Perform the following to verify SSL 2.0 is disabled.

1. If the following key is not present, SSL 2.0 is disabled.

```
HKLM\System\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\SSL 2.0
```

2. Ensure the following key is set to `0`.

```
HKLM\System\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\SSL 2.0\Server\Enabled
```

Remediation:

Perform the following to disable SSL 2.0:

1. If the following key is not present, SSL 2.0 is disabled. You can delete the key to disable the protocol. If you delete the key, steps 2 and 3 are not necessary.

```
HKLM\System\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\SSL 2.0
```

2. If the key exists, set it to `0`.

```
HKLM\System\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\SSL 2.0\Server\Enabled
```

Default Value:

Enabled

References:

1. http://technet.microsoft.com/en-us/library/dn786419.aspx
2. http://technet.microsoft.com/en-us/library/dn786433.aspx
3. http://msdn.microsoft.com/en-us/library/aa374757%28v=vs.85%29.aspx
4. https://www.owasp.org/index.php/Testing_for_SSL-TLS_%28OWASP-CM-001%29

CIS Controls:

14.2 Encrypt All Sensitive Information Over Less-trusted Networks
All communication of sensitive information over less-trusted networks should be encrypted. Whenever information flows over a network with a lower trust level, the information should be encrypted.

7.3 Ensure SSLv3 is disabled (Scored)

Profile Applicability:

- Level 1 - IIS 10

Description:

This protocol is not considered cryptographically secure. Disabling it is recommended.

Rationale:

Disabling weak protocols will help ensure the confidentiality and integrity of in-transit data.

Audit:

Perform the following to verify SSL 3.0 is disabled:

1. Ensure the following key is set to `0`.

```
HKLM\System\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\SS
L 3.0\Server\Enabled
```

Remediation:

Perform the following to disable SSL 3.0:

1. Set the following keyto `0`.

```
HKLM\System\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\SS
L 3.0\Server\Enabled
```

Default Value:

Enabled

References:

1. https://www.openssl.org/~bodo/ssl-poodle.pdf
2. http://technet.microsoft.com/en-us/library/dn786419.aspx
3. https://www.owasp.org/index.php/Testing_for_SSL-TLS_%28OWASP-CM-001%29
4. http://technet.microsoft.com/en-us/library/dn786433.aspx
5. http://msdn.microsoft.com/en-us/library/aa374757%28v=vs.85%29.aspx

CIS Controls:

14.2 Encrypt All Sensitive Information Over Less-trusted Networks
All communication of sensitive information over less-trusted networks should be encrypted. Whenever information flows over a network with a lower trust level, the information should be encrypted.

7.4 Ensure TLS 1.0 is disabled (Not Scored)

Profile Applicability:

- Level 2 - IIS 10

Description:

The PCI Data Security Standard 3.1 recommends disabling "early TLS" along with SSL:

SSL and early TLS are not considered strong cryptography and cannot be used as a security control after June 30, 2016.

Rationale:

This item is Not Scored for the following reasons:

- Enabling TLS 1.2 is recommended.
- These protocols do not suffer from known practical attacks.

Audit:

Review the following registry locations to verify that TLS 1.0 is configured as expected. Disabled settings - Enabled to 0.

```
HKLM\System\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TL
S 1.0\Server\Enabled
```

Remediation:

Set the following registry locations to configure TLS 1.0. To disable, set Enabled to 0.

```
HKLM\System\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TL
S 1.0\Server\Enabled
```

References:

1. http://msdn.microsoft.com/en-us/library/aa374757%28v=vs.85%29.aspx
2. https://www.owasp.org/index.php/Testing_for_SSL-TLS_%28OWASP-CM-001%29
3. http://technet.microsoft.com/en-us/library/dn786419.aspx
4. http://technet.microsoft.com/en-us/library/dn786433.aspx

CIS Controls:

14.2 Encrypt All Sensitive Information Over Less-trusted Networks
All communication of sensitive information over less-trusted networks should be

encrypted. Whenever information flows over a network with a lower trust level, the information should be encrypted.

7.5 Ensure TLS 1.1 is enabled (Not Scored)

Profile Applicability:

- Level 1 - IIS 10

Description:

Enabling TLS 1.1 is required for backward compatibility.

Rationale:

This item is Not Scored for the following reasons:

- Enabling TLS 1.2 is recommended.
- This protocol does not suffer from known practical attacks.

Audit:

Review the following registry locations to verify that TLS 1.1 is enabled. Enabled settings: Enabled to 1.

```
HKLM\System\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TL
S 1.1\Server\Enabled
```

Remediation:

Set the following registry locations to enable TLS 1.1. Set Enabled to 1.

```
HKLM\System\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TL
S 1.1\Server\Enabled
```

References:

1. http://technet.microsoft.com/en-us/library/dn786433.aspx
2. https://www.owasp.org/index.php/Testing_for_SSL-TLS_%28OWASP-CM-001%29
3. http://technet.microsoft.com/en-us/library/dn786419.aspx
4. http://msdn.microsoft.com/en-us/library/aa374757%28v=vs.85%29.aspx

CIS Controls:

14.2 Encrypt All Sensitive Information Over Less-trusted Networks
All communication of sensitive information over less-trusted networks should be encrypted. Whenever information flows over a network with a lower trust level, the information should be encrypted.

7.6 Ensure TLS 1.2 is enabled (Scored)

Profile Applicability:

- Level 1 - IIS 10

Description:

TLS 1.2 is the most recent and mature protocol for protecting the confidentiality and integrity of HTTP traffic. Enabling TLS 1.2 is recommended. This protocol is enabled by default if the registry key is not present. As with any registry changes, a reboot is required for changes to take effect.

Rationale:

Enabling this protocol will help ensure the confidentiality and integrity of data in transit.

Audit:

Perform the following to verify TLS 1.2 has been enabled:

1. Ensure the following key is not present. If it is present, see step 2.

```
HKLM\System\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TL
S 1.2\
```

2. Ensure the following key is set to `0xFFFFFFFF`

```
HKLM\System\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TL
S 1.2\Server\Enabled
```

Remediation:

Perform the following to enable TLS 1.2:

1. Check to see if the following key exists. If it doesn't, TLS 1.2 is enabled by default. If it does, you can delete it or follow step 2.

```
HKLM\System\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TL
S 1.2\
```

2. If the key exists, set the following key to `0xFFFFFFFF`

```
HKLM\System\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TL
S 1.2\Server\Enabled
```

References:

1. http://msdn.microsoft.com/en-us/library/aa374757%28v=vs.85%29.aspx
2. https://www.owasp.org/index.php/Testing_for_SSL-TLS_%28OWASP-CM-001%29
3. http://technet.microsoft.com/en-us/library/dn786419.aspx
4. http://technet.microsoft.com/en-us/library/dn786433.aspx

CIS Controls:

14.2 Encrypt All Sensitive Information Over Less-trusted Networks
All communication of sensitive information over less-trusted networks should be encrypted. Whenever information flows over a network with a lower trust level, the information should be encrypted.

7.7 Ensure NULL Cipher Suites is disabled (Scored)

Profile Applicability:

- Level 1 - IIS 10

Description:

The NULL cipher does not provide data confidentiality or integrity. It is recommended that the NULL cipher be disabled.

Rationale:

By disabling the NULL cipher, there is a better chance of maintaining data confidentiality and integrity.

Audit:

To verify the `NULL` cipher has been disabled, ensure the following key does not exist or is set to `0`:

```
HKLM\System\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Ciphers\NULL
\Enabled
```

Remediation:

To disable the `NULL` cipher, ensure the following key is absent. If the key is present, ensure it is set to `0`.

```
HKLM
System\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Ciphers\NULL\Enab
led
```

References:

1. https://www.owasp.org/index.php/Testing_for_SSL-TLS_%28OWASP-CM-001%29
2. http://technet.microsoft.com/en-us/library/dn786419.aspx
3. http://technet.microsoft.com/en-us/library/dn786433.aspx
4. http://msdn.microsoft.com/en-us/library/aa374757%28v=vs.85%29.aspx

CIS Controls:

14.2 Encrypt All Sensitive Information Over Less-trusted Networks
All communication of sensitive information over less-trusted networks should be encrypted. Whenever information flows over a network with a lower trust level, the information should be encrypted.

7.8 Ensure DES Cipher Suites is disabled (Scored)

Profile Applicability:

- Level 1 - IIS 10

Description:

DES is a weak symmetric-key cipher. It is recommended that it be disabled.

Rationale:

By disabling DES, there is a better chance of maintaining data confidentiality and integrity.

Audit:

To verify the `DES 56/56` cipher has been disabled, ensure the following key does not exist or is set to `0`:

```
HKLM\System\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Ciphers\DES 56/56
```

Remediation:

To disable `DES 56/56`, ensure the following key is absent. If the key is present, ensure it is set to `0`.

```
HKLM\System\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Ciphers\DES 56/56\Enabled
```

References:

1. https://www.owasp.org/index.php/Testing_for_SSL-TLS_%28OWASP-CM-001%29
2. http://technet.microsoft.com/en-us/library/dn786433.aspx
3. http://technet.microsoft.com/en-us/library/dn786419.aspx
4. http://msdn.microsoft.com/en-us/library/aa374757%28v=vs.85%29.aspx

CIS Controls:

14.2 Encrypt All Sensitive Information Over Less-trusted Networks
All communication of sensitive information over less-trusted networks should be encrypted. Whenever information flows over a network with a lower trust level, the information should be encrypted.

7.9 Ensure RC4 Cipher Suites is disabled (Scored)

Profile Applicability:

- Level 1 - IIS 10

Description:

RC4 is a stream cipher that has known practical attacks. It is recommended that RC4 be disabled. The only RC4 cipher enabled by default on Server 2012 and 2012 R2 is RC4 128/128.

Rationale:

The use of RC4 may increase an adversaries ability to read sensitive information sent over SSL/TLS.

Audit:

To verify the `RC4 40/128` cipher has been disabled, ensure the following key does not exist or is set to `0`:

```
HKLM\System\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Ciphers\RC4
40/128\Enabled
```

To verify the `RC4 56/128` cipher has been disabled, ensure the following key does not exist or is set to `0`:

```
HKLM\System\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Ciphers\RC4
56/128\Enabled
```

To verify the `RC4 64/128` cipher has been disabled, ensure the following key does not exist or is set to `0`:

```
HKLM\System\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Ciphers\RC4
64/128\Enabled
```

To verify the `RC4 128/128` cipher has been disabled, ensure the following key is set to `0`:

```
HKLM\System\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Ciphers\RC4
128/128\Enabled
```

Remediation:

To disable `RC4 40/128`, ensure the following key is absent. If the key is present, ensure it is set to `0`.

```
HKLM\System\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Ciphers\RC4
40/128\Enabled
```

To disable `RC4 56/128`, ensure the following key is absent. If the key is present, ensure it is set to `0`.

```
HKLM\System\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Ciphers\RC4
56/128\Enabled
```

To disable `RC4 64/128`, ensure the following key is absent. If the key is present, ensure it is set to `0`.

```
HKLM\System\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Ciphers\RC4
64/128\Enabled
```

To disable `RC4 128/128`, ensure the following key is set to `0`.

```
HKLM\System\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Ciphers\RC4
128/128\Enabled
```

References:

1. http://msdn.microsoft.com/en-us/library/aa374757%28v=vs.85%29.aspx
2. http://technet.microsoft.com/en-us/library/dn786433.aspx
3. http://technet.microsoft.com/en-us/library/dn786419.aspx
4. https://www.owasp.org/index.php/Testing_for_SSL-TLS_%28OWASP-CM-001%29

CIS Controls:

14.2 Encrypt All Sensitive Information Over Less-trusted Networks
All communication of sensitive information over less-trusted networks should be encrypted. Whenever information flows over a network with a lower trust level, the information should be encrypted.

7.10 Ensure Triple DES Cipher Suite is Disabled (Scored)

Profile Applicability:

- Level 1 - IIS 10

Description:

Triple DES Cipher Suites is now considered a weak cipher and is not recommended for use.

Rationale:

This item is Not Scored for the following reasons:

- Enabling AES 256/256 is recommended.
- This cipher does not suffer from known practical attacks.

Audit:

To verify the `Triple DES 168/168` cipher has been disabled, ensure the following key is set to `00000000`:

```
HKLM\System\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Ciphers\Trip
le DES 168/168\Enabled
```

Remediation:

To disable `Triple DES 168/168`, ensure the following key is set to `00000000`.

```
HKLM\System\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Ciphers\Trip
le DES 168/168\Enabled
```

CIS Controls:

14.2 Encrypt All Sensitive Information Over Less-trusted Networks
All communication of sensitive information over less-trusted networks should be encrypted. Whenever information flows over a network with a lower trust level, the information should be encrypted.

7.11 Ensure AES 128/128 Cipher Suite is configured (Not Scored)

Profile Applicability:

- Level 1 - IIS 10

Description:

Enabling AES 128/128 may be required for client compatibility. Enable or disable this cipher suite accordingly.

Rationale:

This item is Not Scored for the following reasons:

- Enabling AES 256/256 is recommended.
- This cipher does not suffer from known practical attacks.

Audit:

To verify the `AES 128/128` cipher has been enabled, ensure the following key is set to `0xFFFFFFF`:

```
HKLM\System\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Ciphers\AES
128/128\Enabled
```

Remediation:

To enable the `AES 128/128` cipher, ensure the following key is set to `0xFFFFFFFF`:

```
HKLM\System\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Ciphers\AES
128/128\Enabled
```

References:

1. http://technet.microsoft.com/en-us/library/dn786419.aspx
2. http://msdn.microsoft.com/en-us/library/aa374757%28v=vs.85%29.aspx
3. https://www.owasp.org/index.php/Testing_for_SSL-TLS_%28OWASP-CM-001%29
4. http://technet.microsoft.com/en-us/library/dn786433.aspx

CIS Controls:

14.2 Encrypt All Sensitive Information Over Less-trusted Networks
All communication of sensitive information over less-trusted networks should be encrypted. Whenever information flows over a network with a lower trust level, the information should be encrypted.

7.12 Ensure AES 256/256 Cipher Suite is enabled (Scored)

Profile Applicability:

- Level 1 - IIS 10

Description:

AES 256/256 is the most recent and mature cipher suite for protecting the confidentiality and integrity of HTTP traffic. Enabling AES 256/256 is recommended. This is enabled by default on Server 2012 and 2012 R2.

Rationale:

Enabling this cipher will help ensure the confidentiality and integrity of data in transit.

Audit:

To verify the `AES 256/256` cipher has been enabled:

1. Ensure that the following key does not exist. If it does exist, you can either delete the key or proceed to step 2.

```
HKLM\System\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Ciphers\AES
256/256\
```

2. If the following key exists, ensure the following is set to 0x`FFFFFFFF`:

```
HKLM\System\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Ciphers\AES
256/256\Enabled
```

Remediation:

To enable the `AES 256/256` cipher:

1. Ensure that the following key does not exist. If it does exist, you can either delete the key or proceed to step 2.

```
HKLM\System\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Ciphers\AES
256/256\
```

2. If the key exists, ensure the following is set to 0x`FFFFFFFF`.

```
HKLM\System\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Ciphers\AES
256/256\Enabled
```

References:

1. http://technet.microsoft.com/en-us/library/dn786419.aspx
2. http://msdn.microsoft.com/en-us/library/aa374757%28v=vs.85%29.aspx
3. https://www.owasp.org/index.php/Testing_for_SSL-TLS_%28OWASP-CM-001%29
4. http://technet.microsoft.com/en-us/library/dn786433.aspx

CIS Controls:

14.2 Encrypt All Sensitive Information Over Less-trusted Networks
All communication of sensitive information over less-trusted networks should be encrypted. Whenever information flows over a network with a lower trust level, the information should be encrypted.

7.13 Ensure TLS Cipher Suite ordering is configured (Scored)

Profile Applicability:

- Level 2 - IIS 10

Description:

Cipher suites are a named combination of authentication, encryption, message authentication code, and key exchange algorithms used for the security settings of a network connection using TLS protocol. Clients send a cipher list and a list of ciphers that it supports in order of preference to a server. The server then replies with the cipher suite that it selects from the client cipher suite list.

Rationale:

Cipher suites should be ordered from strongest to weakest in order to ensure that the more secure configuration is used for encryption between the server and client.

TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384

TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256

TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384

TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256

TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384

TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256

TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384

TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256

Avoid cipher suits that do not provide Perfect Forward Secrecy or use weak hashing function, use them only if you need to support backwards compatibility and in the bottom of the list:

TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA (uses SHA-1)

TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (uses SHA-1)

TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA (uses SHA-1)

TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (uses SHA-1)

TLS_RSA_WITH_AES_256_GCM_SHA384 (lack of Perfect Forward Secrecy)

TLS_RSA_WITH_AES_128_GCM_SHA256 (lack of Perfect Forward Secrecy)

TLS_RSA_WITH_AES_256_CBC_SHA256 (lack of Perfect Forward Secrecy)

TLS_RSA_WITH_AES_128_CBC_SHA256 (lack of Perfect Forward Secrecy)

TLS_RSA_WITH_AES_256_CBC_SHA (uses SHA-1, lack of Perfect Forward Secrecy)

TLS_RSA_WITH_AES_128_CBC_SHA (uses SHA-1, lack of Perfect Forward Secrecy)

Note: HTTP/2 compatibility: first 4 ciphers (in bold) in the top part list are compatible with HTTP/2

Audit:

To verify that the cipher order has been configured properly, navigate to the following registry path.

```
HKLM\System\CurrentControlSet\Control\Cryptography\Configuration\Local\SSL\00
010002\Functions
```

The correct cipher order is listed below

TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384

TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256

TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384

TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256

TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384

TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256

TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384

TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256

Bottom part, backward compatibility list and ordering of ciphers (may be missing or incomplete):

TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA

TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA

TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA

TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA

TLS_RSA_WITH_AES_256_GCM_SHA384

TLS_RSA_WITH_AES_128_GCM_SHA256

TLS_RSA_WITH_AES_256_CBC_SHA256

TLS_RSA_WITH_AES_128_CBC_SHA256

TLS_RSA_WITH_AES_256_CBC_SHA

TLS_RSA_WITH_AES_128_CBC_SHA

Remediation:

To establish the recommended configuration via registry setting, set the following registry key path to and confirm it is set as prescribed.

```
HKLM\System\CurrentControlSet\Control\Cryptography\Configuration\Local\SSL\00
010002\Functions
```

```
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA2
56,TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA25
6,TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384,TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SH
A256,TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384,TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
256
```

OR

To establish the recommended configuration via GP, set the following UI path to

```
Enabled:TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,TLS_ECDHE_ECDSA_WITH_AES_128_
GCM_SHA256,TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384,TLS_ECDHE_RSA_WITH_AES_128_G
CM_SHA256,TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384,TLS_ECDHE_ECDSA_WITH_AES_12
8_CBC_SHA256,TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384,TLS_ECDHE_RSA_WITH_AES_128
_CBC_SHA256
```

```
Computer Configuration\Administrative Templates\Network\SSL Configuration
Settings\SSL Cipher Suite Order
```

This group policy setting is backed by the following registry:

```
HKLM\System\CurrentControlSet\Control\Cryptography\Configuration\Local\SSL\00
01000\:Functions
```

Add backward compatibility ciphers as needed:

```
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA,TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA,TLS_E
CDHE_ECDSA_WITH_AES_128_CBC_SHA,TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA,TLS_RSA_WI
```

```
TH_AES_256_GCM_SHA384,TLS_RSA_WITH_AES_128_GCM_SHA256,TLS_RSA_WITH_AES_256_CB
C_SHA256,TLS_RSA_WITH_AES_128_CBC_SHA256,TLS_RSA_WITH_AES_256_CBC_SHA,TLS_RSA
_WITH_AES_128_CBC_SHA
```

Impact:

Cipher ordering is important to ensure that the most secure ciphers are listed first and will be applied over weaker ciphers when possible.

CIS Controls:

14.2 Encrypt All Sensitive Information Over Less-trusted Networks
All communication of sensitive information over less-trusted networks should be encrypted. Whenever information flows over a network with a lower trust level, the information should be encrypted.

Appendix: Summary Table

Control		Set Correctly	
		Yes	No
1	**Basic Configurations**		
1.1	Ensure web content is on non-system partition (Scored)	☐	☐
1.2	Ensure 'host headers' are on all sites (Scored)	☐	☐
1.3	Ensure 'directory browsing' is set to disabled (Scored)	☐	☐
1.4	Ensure 'application pool identity' is configured for all application pools (Scored)	☐	☐
1.5	Ensure 'unique application pools' is set for sites (Scored)	☐	☐
1.6	Ensure 'application pool identity' is configured for anonymous user identity (Scored)	☐	☐
2	**Configure Authentication and Authorization**		
2.1	Ensure 'global authorization rule' is set to restrict access (Not Scored)	☐	☐
2.2	Ensure access to sensitive site features is restricted to authenticated principals only (Not Scored)	☐	☐
2.3	Ensure 'forms authentication' require SSL (Scored)	☐	☐
2.4	Ensure 'forms authentication' is set to use cookies (Scored)	☐	☐
2.5	Ensure 'cookie protection mode' is configured for forms authentication (Scored)	☐	☐
2.6	Ensure transport layer security for 'basic authentication' is configured (Scored)	☐	☐
2.7	Ensure 'passwordFormat' is not set to clear (Scored)	☐	☐
2.8	Ensure 'credentials' are not stored in configuration files (Scored)	☐	☐
3	**ASP.NET Configuration Recommendations**		
3.1	Ensure 'deployment method retail' is set (Scored)	☐	☐
3.2	Ensure 'debug' is turned off (Scored)	☐	☐
3.3	Ensure custom error messages are not off (Scored)	☐	☐
3.4	Ensure IIS HTTP detailed errors are hidden from displaying remotely (Scored)	☐	☐
3.5	Ensure ASP.NET stack tracing is not enabled (Scored)	☐	☐
3.6	Ensure 'httpcookie' mode is configured for session state (Scored)	☐	☐
3.7	Ensure 'cookies' are set with HttpOnly attribute (Scored)	☐	☐
3.8	Ensure 'MachineKey validation method - .Net 3.5' is configured (Scored)	☐	☐
3.9	Ensure 'MachineKey validation method - .Net 4.5' is configured (Scored)	☐	☐

3.10	Ensure global .NET trust level is configured (Scored)	☐	☐
4	**Request Filtering and Other Restriction Modules**		
4.1	Ensure 'maxAllowedContentLength' is configured (Not Scored)	☐	☐
4.2	Ensure 'maxURL request filter' is configured (Scored)	☐	☐
4.3	Ensure 'MaxQueryString request filter' is configured (Scored)	☐	☐
4.4	Ensure non-ASCII characters in URLs are not allowed (Scored)	☐	☐
4.5	Ensure Double-Encoded requests will be rejected (Scored)	☐	☐
4.6	Ensure 'HTTP Trace Method' is disabled (Scored)	☐	☐
4.7	Ensure Unlisted File Extensions are not allowed (Scored)	☐	☐
4.8	Ensure Handler is not granted Write and Script/Execute (Scored)	☐	☐
4.9	Ensure 'notListedIsapisAllowed' is set to false (Scored)	☐	☐
4.10	Ensure 'notListedCgisAllowed' is set to false (Scored)	☐	☐
4.11	Ensure 'Dynamic IP Address Restrictions' is enabled (Not Scored)	☐	☐
5	**IIS Logging Recommendations**		
5.1	Ensure Default IIS web log location is moved (Scored)	☐	☐
5.2	Ensure Advanced IIS logging is enabled (Scored)	☐	☐
5.3	Ensure 'ETW Logging' is enabled (Not Scored)	☐	☐
6	**FTP Requests**		
6.1	Ensure FTP requests are encrypted (Scored)	☐	☐
6.2	Ensure FTP Logon attempt restrictions is enabled (Not Scored)	☐	☐
7	**Transport Encryption**		
7.1	Ensure HSTS Header is set (Not Scored)	☐	☐
7.2	Ensure SSLv2 is disabled (Scored)	☐	☐
7.3	Ensure SSLv3 is disabled (Scored)	☐	☐
7.4	Ensure TLS 1.0 is disabled (Not Scored)	☐	☐
7.5	Ensure TLS 1.1 is enabled (Not Scored)	☐	☐
7.6	Ensure TLS 1.2 is enabled (Scored)	☐	☐
7.7	Ensure NULL Cipher Suites is disabled (Scored)	☐	☐
7.8	Ensure DES Cipher Suites is disabled (Scored)	☐	☐
7.9	Ensure RC4 Cipher Suites is disabled (Scored)	☐	☐
7.10	Ensure Triple DES Cipher Suite is Disabled (Scored)	☐	☐
7.11	Ensure AES 128/128 Cipher Suite is configured (Not Scored)	☐	☐
7.12	Ensure AES 256/256 Cipher Suite is enabled (Scored)	☐	☐
7.13	Ensure TLS Cipher Suite ordering is configured (Scored)	☐	☐

Appendix: Change History

Date	Version	Changes for this version
3/31/17	1.0.0	Initial Release