Harden IIS

```
Import-Module IISAdministration

Get-IISAppPool
$apppoolname = Read-Host 'What is the ApplicationPoolIdentity name? [default: DefaultAppPool]'

Get-Website | Select-Object Name, PhysicalPath
$websitename = Read-Host 'What is the Website name? [default: Default Web Site]'
$newloglocation = Read-Host 'Set a new IIS web log location other than C: [default:
C:\inetpub\logs\LogFiles\W3SVC1]'

Write-Host "1. Basic Configurations"
Write-Host "  1.1 Ensure 'directory browsing' is set to disabled"
# Ensuring that directory browsing is disabled may reduce the probability of disclosing sensitive content that is
inadvertently accessible via IIS.
Set-WebConfigurationProperty -Filter system.webserver/directorybrowse -PSPath iis:\ -Name Enabled -Value
False

Write-Host "  1.2 Ensure 'Application pool identity' is configured for all application pools"
# 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.
Set-WebConfigurationProperty -pspath 'MACHINE/WEBROOT/APPHOST' -filter
"system.applicationHost/applicationPools/add[@name='$apppoolname']/processModel" -name 'identityType' -
value 'ApplicationPoolIdentity'

Write-Host "  1.3 Ensure 'unique application pools' is set for sites"
# 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.
Set-ItemProperty -Path "IIS:\Sites\$websitename" -Name applicationPool -Value $websitename

Write-Host "  1.4 Ensure 'application pool identity' is configured for anonymous user identity"
# 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.
Set-ItemProperty -Path IIS:\AppPools\$apppoolname -Name passAnonymousToken -Value True

Write-Host " 1.5 Ensure WebDav feature is disabled"
WebDAV is not widely used, and it has serious security concerns because it may allow clients to modify unauthorized files on the web server. Therefore, the WebDav feature should be disabled.
Remove-WindowsFeature Web-DAV-Publishing

Write-Host "2. Configure Authentication and Authorization"
Ensure 'global authorization rule' is set to restrict access
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.
Remove-WebConfigurationProperty -pspath 'MACHINE/WEBROOT/APPHOST' -filter "system.webServer/security/authorization" -name "." -AtElement @{users='*';roles='';verbs=''}
Add-WebConfigurationProperty -pspath 'MACHINE/WEBROOT/APPHOST' -filter "system.webServer/security/authorization" -name "." -value @{accessType='Allow';roles='Administrators'}

Write-Host " 2.1 Ensure access to sensitive site features is restricted to authenticated principals only"
Add the forms tag within <system.web>:
<system.web>
<authentication>
<forms cookieless="UseCookies" requireSSL="true" />
</authentication>
</system.web>
Set-WebConfigurationProperty -pspath 'MACHINE/WEBROOT/APPHOST' -filter 'system.webServer/security/authentication/anonymousAuthentication' -name 'enabled' -value 'True'
Set-WebConfigurationProperty -pspath 'MACHINE/WEBROOT/APPHOST' -filter 'system.webServer/security/authentication/windowsAuthentication' -name 'enabled' -value 'False'

Write-Host " 2.2 Ensure 'forms authentication' requires SSL"
Requiring SSL for Forms Authentication will protect the confidentiality of credentials during the login process, helping mitigate the risk of stolen user information.
Set-WebConfigurationProperty -pspath "MACHINE/WEBROOT/APPHOST/$websitename" -filter 'system.web/authentication/forms' -name 'requireSSL' -value 'True'

Write-Host " 2.3 Ensure 'forms authentication' is set to use cookies"

```
# 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.
Set-WebConfigurationProperty -pspath "MACHINE/WEBROOT/APPHOST/$websitename" -filter 'system.web/authentication/forms' -name 'cookieless' -value 'UseCookies'

Write-Host "  2.4 Ensure 'cookie protection mode' is configured for forms authentication"
# 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.
Set-WebConfigurationProperty -pspath "MACHINE/WEBROOT/APPHOST/$websitename" -filter 'system.web/authentication/forms' -name 'protection' -value 'All'

Write-Host "  2.5 Ensure transport layer security for 'basic authentication' is configured"
# 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.
Set-WebConfigurationProperty -pspath 'MACHINE/WEBROOT/APPHOST' -location $websitename -filter 'system.webServer/security/access' -name 'sslFlags' -value 'Ssl'

Write-Host "  2.6 Ensure 'passwordFormat' is not set to clear"
# Authentication credentials should always be protected to reduce the risk of stolen authentication credentials.
Set-WebConfigurationProperty -pspath "MACHINE/WEBROOT/APPHOST/$websitename" -filter 'system.web/authentication/forms/credentials' -name 'passwordFormat' -value 'SHA1'

Write-Host "  2.7 Ensure 'credentials' are not stored in configuration files"
# 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.
Remove-WebConfigurationProperty -pspath "MACHINE/WEBROOT/APPHOST/$websitename" -filter 'system.web/authentication/forms/credentials' -name '.'

Write-Host "3. ASP.NET Configuration Recommendations"
Write-Host "  3.1 Ensure 'deployment method retail' is set [Manual]"
# 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.
Write-Host "    # Open the machine.config file located in: %systemroot%\Microsoft.NET\Framework\<framework version>\Config"
Write-Host "    # Add the line <deployment retail='true' /> within the <system.web> section:"
Write-Host "      <system.web>"
Write-Host "        <deployment retail="true" />"
```

```
Write-Host "      </system.web>"
Write-Host "    # Do the same for the 'Microsoft.NET\Framework64' directory"

Write-Host "  3.2 Ensure 'debug' is turned off"
# 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.
Set-WebConfigurationProperty -pspath "MACHINE/WEBROOT/APPHOST/$websitename" -filter "system.web/compilation" -name "debug" -value "False"

Write-Host "  3.3 Ensure custom error messages are not off"
# 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.
Set-WebConfigurationProperty -pspath "MACHINE/WEBROOT/APPHOST/$websitename" -filter "system.web/customErrors" -name "mode" -value "Off"

Write-Host "  3.4 Ensure IIS HTTP detailed errors are hidden from displaying remotely"
# 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.
Set-WebConfigurationProperty -pspath "MACHINE/WEBROOT/APPHOST/$websitename" -filter "system.webServer/httpErrors" -name "errorMode" -value "DetailedLocalOnly"

Write-Host "  3.5 Ensure ASP.NET stack tracing is not enabled"
# 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.
Set-WebConfigurationProperty -pspath "MACHINE/WEBROOT/APPHOST/$websitename" -filter "system.web/trace" -name "enabled" -value "False"

Write-Host "  3.6 Ensure 'httpcookie' mode is configured for session state"
# 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.
Write-Host "    # Locate and open the application's web.config file"
Write-Host "    # Add the httpCookies tag within <system.web>:"
Write-Host "      <configuration>"
Write-Host "        <system.web>"
```

```
Write-Host "          <httpCookies httpOnlyCookies='true' />"
Write-Host "        </system.web>"
Write-Host "      </configuration>"

Write-Host "  3.7 Ensure 'MachineKey validation method - .Net 3.5' is configured [HMACSHA256]"
C:\Windows\system32\inetsrv\appcmd set config /commit:WEBROOT /section:machineKey
/validation:HMACSHA256

Write-Host "  3.8 Ensure 'MachineKey validation method - .Net 4.5' is configured [HMACSHA256]"
# 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. 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.
# Use HMACSHA256 encryption for the ASP.NET Machine Key
Set-WebConfigurationProperty -pspath 'MACHINE/WEBROOT' -filter "system.web/machineKey" -name "validation"
-value "HMACSHA256"

Write-Host "  3.9 Ensure global .NET trust level is configured"
Set-WebConfigurationProperty -pspath 'MACHINE/WEBROOT' -filter "system.web/trust" -name "level" -value
"Medium"

Write-Host "  3.10 Ensure X-Powered-By Header is removed"
# While this is not the only way to fingerprint a site through the response headers, it makes it harder and prevents some potential attackers.
Remove-WebConfigurationProperty -pspath 'MACHINE/WEBROOT/APPHOST' -filter
"system.webserver/httpProtocol/customHeaders" -name "." -AtElement @{name='XPowered-By'}

Write-Host "  3.11 Ensure Server Header is removed"
# While this is not the only way to fingerprint a site through the response headers, it makes it harder and prevents some potential attackers. The server header removal directive is a new feature in IIS 10 that can assist in mitigating this risk.
Set-WebConfigurationProperty -pspath 'MACHINE/WEBROOT/APPHOST/' -filter
"system.webServer/security/requestFiltering" -name "removeServerHeader" -value "True"

Write-Host "4. Request Filtering and other Restriction Modules"
Write-Host "  4.1 Ensure 'maxAllowedContentLength' is configured"
# 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.
```

```
Set-WebConfigurationProperty -pspath 'MACHINE/WEBROOT/APPHOST' -filter
"system.webServer/security/requestFiltering/requestLimits" -name "maxAllowedContentLength" -value
30000000

Write-Host "  4.2 Ensure 'maxURL request filter' is configured"
# 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.
Set-WebConfigurationProperty -pspath 'MACHINE/WEBROOT/APPHOST' -filter
"system.webServer/security/requestFiltering/requestLimits" -name "maxUrl" -value 4096

Write-Host "  4.3 Ensure 'MaxQueryString request filter' is configured"
# 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.
Set-WebConfigurationProperty -pspath 'MACHINE/WEBROOT/APPHOST' -filter
"system.webServer/security/requestFiltering/requestLimits" -name "maxQueryString" -value 2048

Write-Host "  4.4 Ensure non-ASCII characters in URLs are not allowed"
# This feature can help defend against canonicalization attacks, reducing the potential attack surface of servers,
sites, and/or applications.
Set-WebConfigurationProperty -pspath 'MACHINE/WEBROOT/APPHOST' -filter
"system.webServer/security/requestFiltering" -name "allowHighBitCharacters" -value "False"

Write-Host "  4.5 Ensure Double-Encoded requests will be rejected"
# This feature will help prevent attacks that rely on URLs that have been crafted to contain double-encoded
request(s).
Set-WebConfigurationProperty -pspath 'MACHINE/WEBROOT/APPHOST' -filter
"system.webServer/security/requestFiltering" -name "allowDoubleEscaping" -value "True"

Write-Host "  4.6 Ensure 'HTTP Trace Method' is disabled"
# 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.
Add-WebConfigurationProperty -pspath 'MACHINE/WEBROOT/APPHOST' -filter
"system.webServer/security/requestFiltering/verbs" -name "." -value @{verb='TRACE';allowed='False'}

Write-Host "  4.7 Ensure Unlisted File Extensions are not allowed"
# Disallowing all but the necessary file extensions can greatly reduce the attack surface of applications and
servers.
# Set the list of allowed extensions (customise to suit your needs)
$SitePath = 'MACHINE/WEBROOT/APPHOST'
$Filter = 'system.webServer/security/requestFiltering/fileExtensions'
```

```
Add-WebConfigurationProperty -pspath $SitePath -filter $Filter -name "." -value
@{fileExtension='.';allowed='True'}
Add-WebConfigurationProperty -pspath $SitePath -filter $Filter -name "." -value
@{fileExtension='.aspx';allowed='True'}
Add-WebConfigurationProperty -pspath $SitePath -filter $Filter -name "." -value
@{fileExtension='.ashx';allowed='True'}
Add-WebConfigurationProperty -pspath $SitePath -filter $Filter -name "." -value
@{fileExtension='.js';allowed='True'}
Add-WebConfigurationProperty -pspath $SitePath -filter $Filter -name "." -value
@{fileExtension='.css';allowed='True'}
Add-WebConfigurationProperty -pspath $SitePath -filter $Filter -name "." -value
@{fileExtension='.json';allowed='True'}
Add-WebConfigurationProperty -pspath $SitePath -filter $Filter -name "." -value
@{fileExtension='.png';allowed='True'}
Add-WebConfigurationProperty -pspath $SitePath -filter $Filter -name "." -value
@{fileExtension='.woff';allowed='True'}
Add-WebConfigurationProperty -pspath $SitePath -filter $Filter -name "." -value
@{fileExtension='.woff2';allowed='True'}
Add-WebConfigurationProperty -pspath $SitePath -filter $Filter -name "." -value
@{fileExtension='.ttf';allowed='True'}
Add-WebConfigurationProperty -pspath $SitePath -filter $Filter -name "." -value
@{fileExtension='.jpg';allowed='True'}
Add-WebConfigurationProperty -pspath $SitePath -filter $Filter -name "." -value
@{fileExtension='.svg';allowed='True'}

Write-Host "  4.8 Ensure Unlisted File Extensions are not allowed (e.g. .config, .backup, .bat)"
Set-WebConfigurationProperty -pspath 'MACHINE/WEBROOT/APPHOST' -filter
"system.webServer/security/requestFiltering/fileExtensions" -name "allowUnlisted" -value "False"

Write-Host "  4.9 Ensure Handler is not granted Write and Script/Execute"
# 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.
Set-WebConfigurationProperty -pspath 'MACHINE/WEBROOT/APPHOST' -filter "system.webServer/handlers" -
name "accessPolicy" -value "Read,Script"

Write-Host "  4.10 Ensure ‘notListedIsapisAllowed’ is set to false"
# Restricting this attribute to false will help prevent potentially malicious ISAPI extensions from being run.
Set-WebConfigurationProperty -pspath 'MACHINE/WEBROOT/APPHOST' -filter
"system.webServer/security/isapiCgiRestriction" -name "notListedIsapisAllowed" -value "False"
```

```
Write-Host "  4.11 Ensure ‘notListedCgisAllowed’ is set to false"
# Restricting this attribute to false will help prevent unlisted CGI extensions, including potentially malicious CGI scripts from being run.
Set-WebConfigurationProperty -pspath 'MACHINE/WEBROOT/APPHOST' -filter "system.webServer/security/isapiCgiRestriction" -name "notListedCgisAllowed" -value "False"

Write-Host "  4.12 Ensure ‘Dynamic IP Address Restrictions’ is enabled"
# 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.
Set-WebConfigurationProperty -pspath 'MACHINE/WEBROOT/APPHOST' -filter "system.webServer/security/dynamicIpSecurity/denyByConcurrentRequests" -name "enabled" -value "True"
# You can customise this value to suit your needs. Start with 5 and adjust as necessary
Set-WebConfigurationProperty -pspath 'MACHINE/WEBROOT/APPHOST' -filter "system.webServer/security/dynamicIpSecurity/denyByConcurrentRequests" -name "maxConcurrentRequests" -value 5

Write-Host "  4.13 Ensure Double-Encoded Requests will be rejected"
Set-WebConfigurationProperty -pspath 'MACHINE/WEBROOT/APPHOST' -filter 'system.webServer/security/requestFiltering' -name 'allowDoubleEscaping' -value 'True'


Write-Host "5. IIS Logging Recommendations"
Write-Host "  5.1 Ensure Default IIS web log location is moved"
# 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).
Set-WebConfigurationProperty -pspath 'MACHINE/WEBROOT/APPHOST' -filter "system.applicationHost/sites/siteDefaults/logFile" -name "directory" -value "$newloglocation"

Write-Host "  5.1 Ensure ETW Logging is enabled"
Set-ItemProperty -Path "IIS:\Sites\$websitename" -Name logfile.logTargetW3C -Value 'File,ETW'

Write-Host "6. FTP Requests"
Write-Host "  6.1 Ensure FTP requests are encrypted"
# 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.
Set-WebConfigurationProperty -pspath 'MACHINE/WEBROOT/APPHOST' -filter "system.applicationHost/sites/siteDefaults/ftpServer/security/ssl" -name "controlChannelPolicy" -value "SslRequire"
```

```
Set-WebConfigurationProperty -pspath 'MACHINE/WEBROOT/APPHOST' -filter
"system.applicationHost/sites/siteDefaults/ftpServer/security/ssl" -name "dataChannelPolicy" -value "SslRequire"

Write-Host "  6.2 Ensure FTP Logon attempt restrictions is enabled"
# 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.
Set-WebConfigurationProperty -pspath 'MACHINE/WEBROOT/APPHOST' -filter
"system.ftpServer/security/authentication/denyByFailure" -name "enabled" -value "True"

Write-Host "7. Transport Encryption"
Write-Host "  7.1 Ensure HSTS Header is set"
# 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.
# To set the HTTP Header at the server level using an AppCmd.exe command, run the
# following command from an elevated command prompt:
C:\Windows\system32\inetsrv\appcmd.exe set config -section:system.webServer/httpProtocol
/+"customHeaders.[name='StrictTransport-Security',value='max-age=31536000; includeSubDomains;
preload']"
# To set the HTTP Header at the Website level using an AppCmd.exe command, run the
#following command from an elevated command prompt:
C:\Windows\system32\inetsrv\appcmd.exe set config "$websitename" -section:system.webServer/httpProtocol
/+"customHeaders.[name='StrictTransport-Security',value='max-age=31536000; includeSubDomains;
preload']"

Write-Host "  7.2 Ensure SSLv2 is disabled"
# Disabling weak protocols will help ensure the confidentiality and integrity of in-transit data. This protocol is not
considered cryptographically secure.
New-Item 'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\SSL 2.0\Server' -
Force | Out-Null
New-Item 'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\SSL 2.0\Client' -
Force | Out-Null
New-ItemProperty -path 'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\SSL
2.0\Server' -name 'Enabled' -value '0' -PropertyType 'DWord' -Force | Out-Null
New-ItemProperty -path 'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\SSL
2.0\Client' -name 'Enabled' -value '0' -PropertyType 'DWord' -Force | Out-Null
New-ItemProperty -path 'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\SSL
```

```
2.0\Server' -name 'DisabledByDefault' -value '1' -PropertyType 'DWord' -Force | Out-Null
New-ItemProperty -path 'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\SSL 2.0\Client' -name 'DisabledByDefault' -value '1' -PropertyType 'DWord' -Force | Out-Null

Write-Host "  7.3 Ensure SSLv3 is Disabled"
# Disabling weak protocols will help ensure the confidentiality and integrity of in-transit data. This protocol is not considered cryptographically secure.
New-Item 'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\SSL 3.0\Server' -Force | Out-Null
New-Item 'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\SSL 3.0\Client' -Force | Out-Null
New-ItemProperty -path 'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\SSL 3.0\Server' -name 'Enabled' -value '0' -PropertyType 'DWord' -Force | Out-Null
New-ItemProperty -path 'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\SSL 3.0\Client' -name 'Enabled' -value '0' -PropertyType 'DWord' -Force | Out-Null
New-ItemProperty -path 'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\SSL 3.0\Server' -name 'DisabledByDefault' -value '1' -PropertyType 'DWord' -Force | Out-Null
New-ItemProperty -path 'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\SSL 3.0\Client' -name 'DisabledByDefault' -value '1' -PropertyType 'DWord' -Force | Out-Null

Write-Host "  7.4 Ensure TLS 1.0 is Disabled"
# 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.
New-Item 'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS 1.0\Server' -Force | Out-Null
New-Item 'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS 1.0\Client' -Force | Out-Null
New-ItemProperty -path 'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS 1.0\Server' -name 'Enabled' -value '0' -PropertyType 'DWord' -Force | Out-Null
New-ItemProperty -path 'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS 1.0\Client' -name 'Enabled' -value '0' -PropertyType 'DWord' -Force | Out-Null
New-ItemProperty -path 'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS 1.0\Server' -name 'DisabledByDefault' -value '1' -PropertyType 'DWord' -Force | Out-Null
New-ItemProperty -path 'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS 1.0\Client' -name 'DisabledByDefault' -value '1' -PropertyType 'DWord' -Force | Out-Null

Write-Host "  7.5 Ensure TLS 1.1 is Disabled"
# 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.
New-Item 'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS 1.1\Server' -
```

```
Force | Out-Null
New-Item 'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS 1.1\Client' -Force | Out-Null
New-ItemProperty -path 'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS 1.1\Server' -name 'Enabled' -value '0' -PropertyType 'DWord' -Force | Out-Null
New-ItemProperty -path 'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS 1.1\Client' -name 'Enabled' -value '0' -PropertyType 'DWord' -Force | Out-Null
New-ItemProperty -path 'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS 1.1\Server' -name 'DisabledByDefault' -value '1' -PropertyType 'DWord' -Force | Out-Null
New-ItemProperty -path 'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS 1.1\Client' -name 'DisabledByDefault' -value '1' -PropertyType 'DWord' -Force | Out-Null

Write-Host "  7.5 Ensure TLS 1.2 is Enabled"
# TLS 1.2 is the most recent and mature protocol for protecting the confidentiality and integrity of HTTP traffic.
New-Item 'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS 1.2\Server' -Force | Out-Null
New-ItemProperty -path 'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS 1.2\Server' -name 'Enabled' -value '1' -PropertyType 'DWord' -Force | Out-Null
New-ItemProperty -path 'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS 1.2\Server' -name 'DisabledByDefault' -value '0' -PropertyType 'DWord' -Force | Out-Null

Write-Host "  7.6 Ensure NULL Cipher Suites is Disabled"
# The NULL cipher does not provide data confidentiality or integrity. It is recommended that the NULL cipher be disabled.
New-Item 'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Ciphers\NULL' -Force | Out-Null
New-ItemProperty -path 'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Ciphers\NULL' -name 'Enabled' -value '0' -PropertyType 'DWord' -Force | Out-Null

Write-Host "  7.7 Ensure DES Cipher Suites is Disabled"
# DES is a weak symmetric-key cipher. It is recommended that it be disabled.
(Get-Item 'HKLM:\').OpenSubKey('SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Ciphers', $true).CreateSubKey('DES 56/56')
New-ItemProperty -path 'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Ciphers\DES 56/56' -name 'Enabled' -value '0' -PropertyType 'DWord' -Force | Out-Null

Write-Host "  7.8 Ensure RC4 Cipher Suites is Disabled"
# 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.
(Get-Item 'HKLM:\').OpenSubKey('SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Ciphers',
```

```
$true).CreateSubKey('RC4 40/128')
New-ItemProperty -path 'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Ciphers\RC4 40/128' -name 'Enabled' -value '0' -PropertyType 'DWord' -Force | Out-Null
(Get-Item 'HKLM:\').OpenSubKey('SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Ciphers', $true).CreateSubKey('RC4 56/128')
New-ItemProperty -path 'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Ciphers\RC4 56/128' -name 'Enabled' -value '0' -PropertyType 'DWord' -Force | Out-Null
(Get-Item 'HKLM:\').OpenSubKey('SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Ciphers', $true).CreateSubKey('RC4 64/128')
New-ItemProperty -path 'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Ciphers\RC4 64/128' -name 'Enabled' -value '0' -PropertyType 'DWord' -Force | Out-Null
(Get-Item 'HKLM:\').OpenSubKey('SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Ciphers', $true).CreateSubKey('RC4 128/128')
New-ItemProperty -path 'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Ciphers\RC4 128/128' -name 'Enabled' -value '0' -PropertyType 'DWord' -Force | Out-Null

Write-Host "  7.9 Ensure AES 128/128 Cipher Suite is Disabled"
# Enabling AES 128/128 may be required for client compatibility. Enable or disable this cipher suite accordingly. Enabling AES 256/256 is recommended as this cipher does not suffer from known practical attacks.
(Get-Item 'HKLM:\').OpenSubKey('SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Ciphers', $true).CreateSubKey('AES 128/128')
New-ItemProperty -path 'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Ciphers\AES 128/128' -name 'Enabled' -value '0' -PropertyType 'DWord' -Force | Out-Null

Write-Host "  7.10 Ensure AES 256/256 Cipher Suite is Enabled"
# 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.
(Get-Item 'HKLM:\').OpenSubKey('SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Ciphers', $true).CreateSubKey('AES 256/256')
New-ItemProperty -path 'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Ciphers\AES 256/256' -name 'Enabled' -value '1' -PropertyType 'DWord' -Force | Out-Null

Write-Host "  7.11 Ensure Triple DES Cipher Suite 168 is Disabled"
(Get-Item 'HKLM:\').OpenSubKey('SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Ciphers', $true).CreateSubKey('Triple DES 168')
New-ItemProperty -path 'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Ciphers\Triple DES 168' -name 'Enabled' -value '0' -PropertyType 'DWord' -Force | Out-Null
(Get-Item 'HKLM:\').OpenSubKey('SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Ciphers', $true).CreateSubKey('Triple DES 168/168')
reg add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Ciphers\Triple
```

```
DES 168/168" /v Enabled /d 0 /t REG_DWORD /f | Out-Null

Write-Host "  7.11 Ensure TLS Cipher Suite Ordering is Configured"
# 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.
# Configure Strong TLS Cipher Suites to support Perfect Forward Secrecy and HTTP/2 support
# Cipher suites should be ordered from strongest to weakest i.e.
# 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
New-Item 'HKLM:\SOFTWARE\Policies\Microsoft\Cryptography\Configuration\SSL\00010002' -Force | Out-Null
New-ItemProperty -path 'HKLM:\SOFTWARE\Policies\Microsoft\Cryptography\Configuration\SSL\00010002' -name 'Functions' -value 'TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256,TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384,TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384,TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384' -PropertyType 'MultiString' -Force | Out-Null

Get-WebBinding -Port * | Format-List bindingInformation

Write-Host "OK."
```

Revision #1
Created 29 October 2024 06:48:37 by aki
Updated 29 October 2024 06:49:31 by aki