**Table of Contents**

[Overview 2](#_Toc70869923)

[Introduction 2](#_Toc70869924)

[Patch Management Workflow 2](#_Toc70869925)

[Step 1: Assess the Environment 2](#_Toc70869926)

[Step 2: Identify Patches 3](#_Toc70869927)

[Step 3: Evaluate and Plan 3](#_Toc70869928)

[Step 4: Deploy the Patch 3](#_Toc70869929)

[Patching Tier Groups 4](#_Toc70869930)

[Install and Configure a Patch Management Solution - Windows Server Update Services (WSUS) 4](#_Toc70869931)

[Patching Timeframes 12](#_Toc70869932)

[Troubleshooting 13](#_Toc70869933)

[Appendix 13](#_Toc70869934)

[Notes 14](#_Toc70869935)

[Related Articles 15](#_Toc70869936)

# Overview

The following documentation covers the steps necessary to implement a heterogeneous patch management best practices initiative. Included in the documentation are efforts to implement Microsoft's Windows Server Update Services (WSUS) feature to implement and administer patch management utilizing group policy and Active Directory for our Windows Servers.

# Introduction

Our current production environment lacks a process to automate patching across our servers. A strategy for establishing and maintaining the latest updates across our server environment regardless of OS version is needed.

Processes are in place to provide security updates that address vulnerabilities when discovered and maintain a steady routine of patching our server infrastructure. Strategies include doing all that IT can to ensure that it has incorporated the right security essentials for patch management best practices and deliver a consistent patching experience.

# Patch Management Workflow

The following workflow will cover the patching process for our Windows server environment. These steps will answer the questions in how we determine success while creating a security plan for the implementation of patch management.

The following workflow describes the steps

## Step 1: Assess the Environment

Gather the current state of the environment and create a baseline.

1. Are there any current threats or vulnerabilities in your environment?
2. Has anything changed in production?
   1. For example, have you installed new operating systems and applications or made changes to your network or management infrastructure?
3. Do you have accurate and up-to-date inventory information?
4. Is the infrastructure able to support patch management?
5. Are there any inaccessible locations?
6. Firewall rules blocking updates?
7. How are we handling unmanaged devices on the network?
8. Assess and identify assets to be targeted for updates.

* Get a list of all assets and the organizational units (OU) where they reside.
* Identify server operating system version.

1. Group and assign a priority number to assets.

* For example, group domain controllers separate them from other non-domain controllers.
* Use security groups to separate servers based on the value of the asset.
* Determine the cost to maintain the asset.
* Determine the cost of the asset if taken offline.
* Overall, the priority value should reflect asset impairment if taken offline.

1. Machines that cannot receive policy (VM sessions)?
2. Determine the updates deployed since the last deployment.

## Step 2: Identify Patches

Identify the patches to be deployed and the assets that need them.

1. How are we notified about the patches?
2. Identify the new patches.
3. Are the patches relevant to our environment?

* Zero-day
* Critical, Important/High, Moderate, Low
* Bugfix

1. Which systems require patching?
2. Do all your systems need to be patched with the same priority level?
   1. Does each group need to be patched to the same level? (Tier 1 DC, Tier 2 DC, etc.…)?

* Servers exist in three different AD security groups: Tier 3, Tier 2, Tier 1 based on the importance of each server.
* For example, file servers are Tier 3, while domain controllers are considered Tier 1 or 2.
* Different tiers determine patching priority beginning with Tier 3, Tier 2,  then Tier 1.

## Step 3: Evaluate and Plan

Evaluate and verify that the patch works effectively in your environment. Plan the patch release process, get the required sign-offs to deploy the patch and communicate as necessary the timing and impact of the patch deployment.

1. Which of your systems are most vulnerable?

* Domain controllers are most vulnerable, so they should be patched ASAP when updates are made available and patched last.
* On a case-by-case basis, the most vulnerable systems may need more patching than the size of the monthly patch deployment.

1. Have you downloaded the patch?
2. Who is the source of truth?
3. How do we validate updates?

* Do we know that the patch is installed and working correctly?

1. Can the patch be uninstalled?
2. What is the change management process to ensure all stakeholders are aware of the patch(es) installed?
3. Perform risk assessment

* When are you going to install the patch?
* Take into account any outages or 'freeze' periods that may delay deployment

1. Obtain approval to deploy updates
2. Send out communication and roll out updates

## Step 4: Deploy the Patch

Distribute and install the patch on the target systems, review the progress of the deployment, handle any issues, and verify the installation on the target systems.

Prepare production environment for new patches:

1. Designated administrators and users report downtime.
2. Determine the amount of downtime needed for patching our servers.

* Limit the number of servers to be patched at any given instance to reduce the amount of downtime needed.
* The patching time limit minimizes LOE for engineers and eventually users.
* Give yourself enough downtime to account for any issues that may occur.
* For example, after-hours patching may have 10 - 12 servers per patching instance. Allow a 4 - 5 hour window to complete this task.
* The size of the patch can be a significant issue for or distributing the patch across the network and determining the patching time window.
* Caching is not an option at this time.

1. Check distribution points to confirm the presence of the patch.
2. Update synchronization with your source of truth. It should be automatic daily.
3. Patch deployment is NOT automatic. Patching needs approving through our testing process.
4. Are you able to monitor the patch distribution, check progress, and deal with exceptions?

* Use available methods to monitor, check progress and deal with exceptions
* Exceptions are on a case-by-case basis. (i.e., Tableau)

1. What is the rollback plan?

# Patching Tier Groups

# Install and Configure a Patch Management Solution - Windows Server Update Services (WSUS)

The following steps took place on Windows Server 2016 Standard.

NOTE: This step falls outside the scope of configuring WSUS and is not included in the automation process.

1. Verify that you have created an additional hard disk for your VM with 40Gb of space.
2. Initialize the unallocated drive space using NTFS.
3. Select the E: as the drive letter.

The following script will run and configure settings within the Options section of the WSUS console. Not all the options will be configured—just the necessary ones.

1. Open Powershell as Administrator and run the following commands.
2. Install WSUS with Powershell. Using Powershell saves time by avoiding the manual process of adding WSUS through Server Manager.

**PART I**

# Install WSUS with Powershell. Using Powershell saves time by avoiding the manual process of adding WSUS through Server Manager.

Install-WindowsFeature -name UpdateServices -IncludeManagementTools

# Create a directory where updates will be stored.

New-Item -Path E:\WSUS -ItemType Directory

# Change directory to \Update Services\Tools and run wsusutil.exe.

cd "C:\Program Files\Update Services\Tools"

.\wsusutil.exe postinstall CONTENT\_DIR=E:\WSUS

# Inspect the WSUS installation state using best practices analyzer.

Invoke-BpaModel -ModelId Microsoft/Windows/UpdateServices

# Run a scan to determine if you are in compliance with best practices.

Get-BpaResult -ModelId Microsoft/Windows/UpdateServices | Select Title,Severity,Compliance | Format-List

**PART II**

# Finish configuring the WSUS server by synchronizing categories, classifications, and available updates. Get WSUS Server Object.

$wsus = Get-WSUSServer

# Connect to WSUS server configuration.

$wsusConfig = $wsus.GetConfiguration()

# Set to download updates from Microsoft update

Set-WsusServerSynchronization -SyncFromMU

# Set update languages to English and Save Configuration settings.

$wsusConfig.AllUpdateLanguagesEnabled = $false

$wsusConfig.SetEnabledUpdateLanguages("en")

$wsusConfig.Save()

# Get WSUS Subscription and perform initial synchronization to get latest categories

$subscription = $wsus.GetSubscription()

$subscription.StartSynchronizationForCategoryOnly()

# Synchronize with Microsoft Updates

# All of the synchronization steps below may take up to 5 - 10 minutes to finish. This can be done at a later time.

While ($subscription.GetSynchronizationStatus() -ne 'Not Processing') {

Write-Host "." -NoNewline

Start-Sleep -Seconds 5

}

Write-Host "Sync is done."

# Configure the Platforms that we want WSUS to receive updates. Due to size constraints, could be typed on one line.

Get-WsusProduct | Where-Object {$\_.Product.Title -in (

'CAPICOM',

'Silverlight',

'SQL Server 2008 R2',

'SQL Server 2008',

'Windows Server 2008',

'Windows Server 2008 R2',

'Windows Server 2012',

'Windows Server 2012 R2',

'Windows Server 2016')

} | Set-WsusProduct

# Configure the Classifications. Due to size constraints, could be typed on one line.

Get-WsusClassification | Where-Object {$\_.Classification.Title -in (

'Update Rollups',

'Security Updates',

'Critical Updates',

'Service Packs',

'Updates')

} | Set-WsusClassification

# Configure Synchronizations

$subscription.SynchronizeAutomatically = $true

# Set synchronization scheduled for midnight each night

$subscription.SynchronizeAutomaticallyTimeOfDay = (New-TimeSpan -Hours 0)

$subscription.NumberOfSynchronizationsPerDay = 1

$subscription.Save()

# Kick off a synchronization

$subscription.StartSynchronization()

# Create WSUS computer groups. Each computer group should represent each Tier group that we have created for patching priority.

$wsus.CreateComputerTargetGroup("Patching - Tier 0")

$wsus.CreateComputerTargetGroup("Patching - Tier 1")

$wsus.CreateComputerTargetGroup("Patching - Tier 2")

$wsus.CreateComputerTargetGroup("Patching - Tier 3a")

$wsus.CreateComputerTargetGroup("Patching - Tier 3b")

$wsus.CreateComputerTargetGroup("Patching - Tier 3c")

$wsus.CreateComputerTargetGroup("Patching - Tier 3d")

$wsus.CreateComputerTargetGroup("Patching - Tier 3e")

**PART III - WSUS Console and approving updates**

**At this point it is assumed that group policy created for the patching groups is deployed and in effect.**

1. Open the WSUS Console and go to Updates
2. Take note that automatic approval for updates is turned OFF by default.
3. Go to the 'Computers' section and select any "Patching - Tier \*" group. If any server needs updates, it will show up in the Tier group with a yellow warning icon next to it.
4. If a server exists needing updates, highlight the server.
5. Down below, verify if any updates are needed.
6. Select 'Updates needed'. A report will open for the specified server. Move to the third page of the report.
7. Select the update you want to approve. Select 'Not approved.'
8. Highlight the Computer group that should receive the update then click OK.  
   NOTE: The same update can be approved for other computer groups.
9. Once a computer group has all of the chosen updates, the server(s) will receive the updates based on their group policy scheduled time.

|  |  |  |
| --- | --- | --- |
| **Configure Automatic Updates** | **Enabled**  **Configure automatic updating:**  **3 - Auto download and notify for install**  **Scheduled install day:**  **0 - Every day**  **Scheduled install time: 06:00** | **3 = (Default setting) Download the updates automatically and notify when they are ready to be installed** |

1. To verify which group policy is associated to a server, query Active Directory for the server name then verify group membership.
2. The group policy used for patching has 'Client-side Targeting' enabled. A group policy will be created for each "Patching - Tier \*" computer group created. This is a complicated approach, but this is to ensure that only the specified "Patching - Tier \*" computer group within the group policy is targeted. Security filtering is utilized within each group policy so that when the policy is applied to an OU, only members of each patching AD security group is affected. Each AD security group created will correspond with its GPO of the same naming convention.
3. The table below represents the policy used in each GPO. Under the Policy column, second row, there are two references for domain controllers and non-domain controllers.

**Windows Group Policy (GPO)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Settings** | **Supported OS** | **Policy** | **Description** |
| Always automatically restart at the scheduled time | At least windows Server 2012, Windows 8 or Windows RT | Enabled  The restart timer will give users this much time to save their work (minutes): 15 | If you enable this policy, a restart timer will always begin immediately after Windows Update installs important updates, instead of first notifying users on the login screen for at least two days.  The restart timer can be configured to start with any value from 15 to 180 minutes. When the timer runs out, the restart will proceed even if the PC has signed-in users.  If you disable or do not configure this policy, Windows Update will not alter its restart behavior.  If the "No auto-restart with logged on users for scheduled automatic updates installations" policy is enabled, then this policy has no effect. |
| Configure Automatic Updates | Windows XP Professional Service Pack 1 or At least Windows 2000 Service Pack 3 | Enabled  Configure automatic updating:  **For Domain Controllers:**  3 - Auto download and notify for install  **For non-Domain Controllers:**  4 = Automatically download updates and install them on the schedule specified below.  The following settings are only required and applicable if 4 is selected.  Scheduled install day:  0 - Every day  Scheduled install time: 06:00 | Specifies whether this computer will receive security updates and other important downloads through the Windows automatic updating service.  Note: This policy does not apply to Windows RT.  This setting lets you specify whether automatic updates are enabled on this computer. If the service is enabled, you must select one of the four options in the Group Policy Setting:  2 = Notify before downloading and installing any updates.  When Windows finds updates that apply to this computer, users will be notified that updates are ready to be downloaded. After going to Windows Update, users can download and install any available updates.  3 = (Default setting) Download the updates automatically and notify when they are ready to be installed  Windows finds updates that apply to the computer and downloads them in the background (the user is not notified or interrupted during this process). When the downloads are complete, users will be notified that they are ready to install. After going to Windows Update, users can install them.  4 = Automatically download updates and install them on the schedule specified below.  Specify the schedule using the options in the Group Policy Setting. If no schedule is specified, the default schedule for all installations will be every day at 3:00 AM. If any updates require a restart to complete the installation, Windows will restart the computer automatically. (If a user is signed in to the computer when Windows is ready to restart, the user will be notified and given the option to delay the restart.)  On Windows 8 and later, you can set updates to install during automatic maintenance instead of a specific schedule. Automatic maintenance will install updates when the computer is not in use, and avoid doing so when the computer is running on battery power. If automatic maintenance is unable to install updates for 2 days, Windows Update will install updates right away. Users will then be notified about an upcoming restart, and that restart will only take place if there is no potential for accidental data loss.  Automatic maintenance can be further configured by using Group Policy settings here: Computer Configuration->Administrative Templates->Windows Components->Maintenance Scheduler  5 = Allow local administrators to select the configuration mode that Automatic Updates should notify and install updates.  With this option, local administrators will be allowed to use the Windows Update control panel to select a configuration option of their choice. Local administrators will not be allowed to disable the configuration for Automatic Updates.  If the status for this policy is set to Disabled, any updates that are available on Windows Update must be downloaded and installed manually. To do this, search for Windows Update using Start.  If the status is set to Not Configured, use of Automatic Updates is not specified at the Group Policy level. However, an administrator can still configure Automatic Updates through Control Panel. |
| Specify intranet Microsoft update service location | At least Windows XP Professional Service Pack 1 or Windows 2000 Service Pack 3, excluding Windows RT | Enabled  Set the intranet update service for detecting updates: [*<add*](http://lv7-wsus1.test.sandbox.net:8530/) *http path to test server>*  Set the intranet statistics server: [*<add*](http://lv7-wsus1.test.sandbox.net:8530/) *http path to test server>* | Specifies an intranet server to host updates from Microsoft Update. You can then use this update service to automatically update computers on your network.  This setting lets you specify a server on your network to function as an internal update service. The Automatic Updates client will search this service for updates that apply to the computers on your network.  To use this setting, you must set two server name values: the server from which the Automatic Updates client detects and downloads updates, and the server to which updated workstations upload statistics. You can set both values to be the same server. An optional server name value can be specified to configure Windows Update Agent to download updates from an alternate download server instead of the intranet update service.  If the status is set to Enabled, the Automatic Updates client connects to the specified intranet Microsoft update service (or alternate download server), instead of Windows Update, to search for and download updates. Enabling this setting means that end users in your organization don't have to go through a firewall to get updates, and it gives you the opportunity to test updates before deploying them.  If the status is set to Disabled or Not Configured, and if Automatic Updates is not disabled by policy or user preference, the Automatic Updates client connects directly to the Windows Update site on the Internet.  The alternate download server configures the Windows Update Agent to download files from an alternative download server instead of the intranet update service.  The option to download files with missing Urls allows content to be downloaded from the Alternate Download Server when there are no download Urls for files in the update metadata. This option should only be used when the intranet update service does not provide download Urls in the update metadata for files which are present on the alternate download server.  Note: If the "Configure Automatic Updates" policy is disabled, then this policy has no effect.  Note: If the "Alternate Download Server" is not set, it will use the intranet update service by default to download updates.  Note: The option to "Download files with no Url..." is only used if the "Alternate Download Server" is set.  Note: This policy is not supported on Windows RT. Setting this policy will not have any effect on Windows RT PCs. |
| Automatic Updates detection frequency | At least Windows XP Professional Service Pack 1 or Windows 2000 Service Pack 3, excluding Windows RT | Enabled  Check for updates at the following interval (hours): 1 | Specifies the hours that Windows will use to determine how long to wait before checking for available updates. The exact wait time is determined by using the hours specified here minus zero to twenty percent of the hours specified. For example, if this policy is used to specify a 20 hour detection frequency, then all clients to which this policy is applied will check for updates anywhere between 16 and 20 hours.  If the status is set to Enabled, Windows will check for available updates at the specified interval.  If the status is set to Disabled or Not Configured, Windows will check for available updates at the default interval of 22 hours.  Note: The "Specify intranet Microsoft update service location" setting must be enabled for this policy to have effect.  Note: If the "Configure Automatic Updates" policy is disabled, this policy has no effect.  Note: This policy is not supported on Windows RT. Setting this policy will not have any effect on Windows RT PCs. |
| Do not connect to any Windows Update Internet locations | At least Windows Server 2012 R2, Windows 8.1 or Windows RT 8.1 | Enabled | Even when Windows Update is configured to receive updates from an intranet update service, it will periodically retrieve information from the public Windows Update service to enable future connections to Windows Update, and other services like Microsoft Update or the Windows Store.  Enabling this policy will disable that functionality, and may cause connection to public services such as the Windows Store to stop working.  Note: This policy applies only when this PC is configured to connect to an intranet update service using the "Specify intranet Microsoft update service location" policy. |
| Do not include drivers with Windows Updates | At least Windows Server 2016 or Windows 10 | Enabled | Enable this policy to not include drivers with Windows quality updates.  If you disable or do not configure this policy, Windows Update will include updates that have a Driver classification. |
| Enable client-side targeting | At least Windows XP Professional Service Pack 1 or Windows 2000 Service Pack 3, excluding Windows RT | Enabled  Target group name for this computer  Patching - Tier 1  Patching - Tier 2 | Specifies the target group name or names that should be used to receive updates from an intranet Microsoft update service.  If the status is set to Enabled, the specified target group information is sent to the intranet Microsoft update service which uses it to determine which updates should be deployed to this computer.  If the intranet Microsoft update service supports multiple target groups this policy can specify multiple group names separated by semicolons. Otherwise, a single group must be specified.  If the status is set to Disabled or Not Configured, no target group information will be sent to the intranet Microsoft update service.  Note: This policy applies only when the intranet Microsoft update service this computer is directed to is configured to support client-side targeting. If the "Specify intranet Microsoft update service location" policy is disabled or not configured, this policy has no effect. Note: This policy is not supported on Windows RT. Setting this policy will not have any effect on Windows RT PCs. |

# Patching Timeframes

Microsoft has issued the following patch timeframe guidelines. Factors that can impact the release time frames, include:

|  |  |  |
| --- | --- | --- |
| **Severity Rating** | **Definition** | **Patching Timeframe** |
| **Critical** | Remote Code Execution: Any vulnerability which could allow an attacker to execute malicious code on a system without user interaction.  Examples:  Code execution on Hyper-V host from Guest Virtual Machine  Chakra Remote Code Execution  SMB Remote Code Execution | Within 24 hours |
| **Important** | Elevation of Privilege: A vulnerability which allows a low privileged user to bypass controls and operate as a higher privileged user.  Examples:  Win32k Use-after-Free  App container escapes  Windows Defender Application Guard (WDAG) escapes  SOP Bypass vulnerabilities  Escalation from non-administrative user to SYSTEM privileges  Information Disclosure: A vulnerability which allows an attacker to obtain access to data which should be protected during normal operation.  Examples:  Uninitialized kernel memory disclosure to user mode  Remote Code Execution: Any vulnerability which could allow execution of malicious code but requires user interaction.  Examples:  Heap buffer overrun via a document file  Denial of Service: A vulnerability which allows an attacker to disrupt the system and interrupt or halt normal operations.  Examples:  Remotely triggerable resource exhaustion issues  Remotely triggerable issues that result in a reboot  Security Feature Bypass: A vulnerability which allows an attacker to circumvent controls or features designed to protect users.  Examples:  WDSG bypasses  BitLocker bypasses | Within one month |
| **Moderate** | Denial of Service: A vulnerability which allows an attacker to disrupt the system and interrupt or halt normal operations.  Examples:  Locally triggerable with no remote vector  Remotely triggerable but requiring many attackers (DDoS)  Information Disclosure: A vulnerability which allows an attacker to obtain access to data which should be protected during normal operation. Examples:  Kernel pool pointer disclosure | Wait for next service pack or patch rollup that includes the patch, or deploy the patch within 4 months |
| **Low** | Exploitation is extremely difficult or impact is minimal | Wait for next service pack or patch rollup that includes the patch, or deploy the patch within 1 year |

# Troubleshooting

As setup is involved with most areas of IT, things will break, so troubleshooting is just as important.

# Appendix

The following components are installed when configuring WSUS through Powershell.

|  |  |  |
| --- | --- | --- |
| "[Windows Server Update Services] Windows Server Update Services" | "[Windows Server Update Services] WID Connectivity" | "[Windows Server Update Services] WSUS Services" |
| "[Remote Server Administration Tools] API and PowerShell cmdlets" | "[Remote Server Administration Tools] Windows Server Update Services Tools" | "[Remote Server Administration Tools] Role Administration Tools" |
| "[Remote Server Administration Tools] Remote Server Administration Tools" | "[Web Server (IIS)] IIS 6 Metabase Compatibility" | "[Web Server (IIS)] IIS 6 Management Compatibility" |
| "[Web Server (IIS)] Management Tools" | "[Web Server (IIS)] Web Server (IIS)" | "[Web Server (IIS)] IIS Management Console" |
| "[Web Server (IIS)] Windows Authentication" | "[Web Server (IIS)] Security" | "[Web Server (IIS)] Web Server" |
| "[Web Server (IIS)] ASP.NET 4.6" | "[Web Server (IIS)] Application Development" | "[Web Server (IIS)] Request Filtering" |
| "[Web Server (IIS)] Default Document" | "[Web Server (IIS)] Common HTTP Features" | "[Web Server (IIS)] ISAPI Filters" |
| "[Web Server (IIS)] ISAPI Extensions" | "[Web Server (IIS)] .NET Extensibility 4.6" | "[.NET Framework 4.6 Features] ASP.NET 4.6" |
| "[Web Server (IIS)] Dynamic Content Compression" | "[Web Server (IIS)] Performance" | "[Windows Process Activation Service] Configuration APIs" |
| "[Windows Process Activation Service] Windows Process Activation Service" | "[Web Server (IIS)] Static Content" | "[.NET Framework 4.6 Features] HTTP Activation" |
| "[Windows Process Activation Service] Process Model" | "[Windows Internal Database] Windows Internal Database" | "[Remote Server Administration Tools] User Interface Management Console" |

The following completion message is displayed:

The target server may need to be restarted after the installation completes.

Success Restart Needed Exit Code Feature Result

------- -------------- --------- --------------   
True Maybe Success {Windows Server Update Services, WID Conne...

Performing installation for "[Windows Internal Database] Windows Internal Database".

Windows Internal Database (WID) ships with the OS. No additional licensing needed.

Relational database stored here: C:\Windows\WID\Data\SUSDB.mdf

# Notes

* Subscribe to notification services or Automate notification services
* RESTful API can be used to engage the Microsoft Security Response Center (MSRC) in the following ways:<https://github.com/Microsoft/MSRC-Microsoft-Security-Updates-API/blob/master/src/README.md>

|  |
| --- |
| ### Install the module from the PowerShell Gallery (must be run as Admin)  Install-Module -Name msrcsecurityupdates -force  Import-module msrcsecurityupdates  Set-MSRCApiKey -ApiKey "insert API key here" -Verbose  $monthOfInterest = '2018-Oct'  Get-MsrcCvrfDocument -ID $monthOfInterest -Verbose | Get-MsrcSecurityBulletinHtml -Verbose | Out-File c:\windows\temp\MSRCOctober2018SecurityUpdates.html |

* Report suspected cyber attacks or abuse originating from Microsoft Online ServicesGet security update summaries and details using the Common Vulnerability Reporting Framework - <https://www.icasi.org/cvrf>
* Notify Microsoft of any planned penetration tests against your Azure assets (if we had running instances in Azure)
* Microsoft security guidance websites
  1. <https://portal.msrc.microsoft.com/en-us/security-guidance>
* Product-specific pages
  1. <https://docs.microsoft.com/en-us/windows-server/administration/windows-server-update-services/get-started/windows-server-update-services-wsus>
* Third-party sites - WSUS does not provide third-party patching.
* Implement regular review and deployment schedule
* Microsoft's patch release schedule second Tuesday of each month
* Configure/Verify automated tools to check for new updates daily

# Related Articles

<https://portal.msrc.microsoft.com/en-us/security-guidance>