

.Net7 data to remediate

```
powershell.exe -NoProfile -ExecutionPolicy Bypass -File Remove-Obsolete-AspNetCore.ps1 -WhatIfOnly
```

```
powershell.exe -NoProfile -ExecutionPolicy Bypass -File Remove-Obsolete-AspNetCore.ps1
```

How to wire this into Tanium:

1. Detection package (recommended first, deploy to a test group):

- Command line: `powershell.exe -NoProfile -ExecutionPolicy Bypass -File Remove-Obsolete-AspNetCore.ps1 -WhatIfOnly`
- Files: attach the `.ps1`
- Check the resulting log at `%WINDIR%\Temp\tanium\Remove-Obsolete-AspNetCore.log` on a few pilot endpoints — confirm the "WOULD REMOVE" lines line up with what you expect and nothing gets skipped that you actually wanted gone.

2. Remediation package:

- Command line: `powershell.exe -NoProfile -ExecutionPolicy Bypass -File Remove-Obsolete-AspNetCore.ps1`
- Exit code result: 0 = success, 2 = success/reboot required, 1603 = one or more removals failed, 99 = script error. Set your Tanium action's success criteria on exit code 0 or 2.
- Timeout: give it a couple minutes per endpoint — sequential `msiexec /x` calls aren't instant, especially with several branches queued.

3. Verification sensor — reissue your original Comply/vuln check, or a quick custom sensor querying the same uninstall registry paths filtered on those DisplayVersion strings, to confirm they're gone post-action.

A few things you'll probably want to adjust before rollout:

- The default `$VersionsToRemove` list is hardcoded to the six versions from your scan — update it each patch cycle, or better, pass it in via a Tanium parameterized package so you're not editing the script every month.
- The branch-safety check assumes a newer build of the *same* major.minor is already present. If a machine only has the vulnerable build and nothing newer (e.g., patch didn't land yet), it'll skip and log — worth alerting on those skips so you catch machines that need the actual runtime update pushed first.

```
<#
.SYNOPSIS
    Removes obsolete/vulnerable ASP.NET Core and .NET Runtime versions, intended
    for deployment as a Tanium package.

.DESCRIPTION
    Scans both 64-bit and WOW6432Node uninstall registry hives for ASP.NET Core /
    .NET Runtime / .NET Host FX Resolver entries. For each entry whose DisplayVersion
    matches the -VersionsToRemove list, it will only uninstall if a NEWER version of
    the SAME major.minor branch is already installed (side-by-side safety check).
    This prevents removing the only runtime an app depends on.

    Designed to be idempotent - safe to re-run. Entries not found are simply skipped.

.PARAMETER VersionsToRemove
    Exact DisplayVersion strings to target for removal (as reported by the vuln scan).

.PARAMETER WhatIfOnly
    If specified, only reports what WOULD be removed. No changes made. Use this as
    a detection-only Tanium package/sensor before rolling out the live action.

.PARAMETER LogPath
    Where to write the log file. Defaults to Tanium's own working temp dir.

.EXAMPLE
    powershell.exe -NoProfile -ExecutionPolicy Bypass -File Remove-Obsolete-AspNetCore.ps1

.EXAMPLE
    powershell.exe -NoProfile -ExecutionPolicy Bypass -File Remove-Obsolete-AspNetCore.ps1 -
    WhatIfOnly

.NOTES
    Exit codes (Tanium-friendly):
        0    = Success (removed target versions, or nothing matched - nothing to do)
        2    = Success, reboot required (maps from msiexec 3010)
        1603 = One or more msiexec calls failed - check log
        99   = Unexpected script error
#>

[CmdletBinding()]
```

```

param(
    [string[]]$VersionsToRemove = @(
        '2.1.30.60071',
        '6.0.36.24516',
        '7.0.20.24269',
        '8.0.14.25112',
        '9.0.3.25112',
        '9.0.9.25420'
    ),
    [switch]$WhatIfOnly,
    [string]$LogPath = "$env:WINDIR\Temp\tanium\Remove-Obsolete-AspNetCore.log"
)

$erroractionpreference = 'Stop'
$exitCode = 0
$rebootRequired = $false
$failures = 0

New-Item -ItemType Directory -Path (Split-Path $LogPath) -Force | Out-Null

function Write-Log {
    param([string]$Message)
    $line = "[{0}] {1}" -f (Get-Date -Format 's'), $Message
    $line | Tee-Object -FilePath $LogPath -Append | Out-Null
    Write-Host $line
}

function Get-MajorMinor {
    param([string]$Version)
    $parts = $Version -split '\.'
    if ($parts.Count -ge 2) { return "$($parts[0]).$($parts[1])" }
    return $Version
}

Write-Log "=== Remove-Obsolete-AspNetCore starting (WhatIfOnly=$WhatIfOnly) ==="
Write-Log "Target versions: $($VersionsToRemove -join ', ')"

try {
    $uninstallRoots = @(
        'HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall',

```

```

        'HKLM:\SOFTWARE\WOW6432Node\Microsoft\Windows\CurrentVersion\Uninstall'
    )

    $namePattern = 'ASP\.NET Core|Microsoft \.NET Runtime|\.NET Host FX Resolver|Microsoft
\ .NET Host'

    $allEntries = foreach ($root in $uninstallRoots) {
        if (-not (Test-Path $root)) { continue }
        Get-ChildItem $root | ForEach-Object {
            $props = Get-ItemProperty -Path $_.PSPath -ErrorAction SilentlyContinue
            if ($props.DisplayName -and $props.DisplayName -match $namePattern -and
$props.DisplayVersion) {
                [PSCustomObject]@{
                    ProductCode   = $_.PSChildName
                    DisplayName    = $props.DisplayName
                    DisplayVersion = $props.DisplayVersion
                    MajorMinor     = Get-MajorMinor $props.DisplayVersion
                    RegistryPath   = $_.PSPath
                }
            }
        }
    }

    if (-not $allEntries) {
        Write-Log "No matching ASP.NET Core / .NET Runtime entries found on this endpoint."
        Write-Log "=== Complete. Exit code: 0 ==="
        exit 0
    }

    # Determine the highest installed version per major.minor branch, per DisplayName family
    $branchMax = @{}
    foreach ($entry in $allEntries) {
        $key = "$($entry.DisplayName -replace '\d+(\.\d+){1,3}.*$', '')|$($entry.MajorMinor)"
        $current = $branchMax[$key]
        if (-not $current -or ([version]$entry.DisplayVersion -gt [version]$current)) {
            $branchMax[$key] = $entry.DisplayVersion
        }
    }

    foreach ($entry in $allEntries) {

```

```

if ($entry.DisplayVersion -notin $VersionsToRemove) { continue }

$key = "$($entry.DisplayName -replace '\d+(\.\d+){1,3}.*$', '')| $($entry.MajorMinor)"
$maxInBranch = $branchMax[$key]

if ([version]$entry.DisplayVersion -ge [version]$maxInBranch) {
    Write-Log "SKIP (no newer build present) - $($entry.DisplayName)
[$($entry.DisplayVersion)]"
    continue
}

if ($WhatIfOnly) {
    Write-Log "WOULD REMOVE - $($entry.DisplayName) [$($entry.DisplayVersion)]
ProductCode=$($entry.ProductCode)"
    continue
}

Write-Log "REMOVING - $($entry.DisplayName) [$($entry.DisplayVersion)]
ProductCode=$($entry.ProductCode)"
$msiArgs = "/x $($entry.ProductCode) /quiet /norestart /l*v
`$env:WINDIR\Temp\tanium\msi_$( $entry.ProductCode ).log`"
$proc = Start-Process -FilePath msixec.exe -ArgumentList $msiArgs -Wait -PassThru -
WindowStyle Hidden

switch ($proc.ExitCode) {
    0 { Write-Log " -> Success" }
    3010 { Write-Log " -> Success, reboot required"; $rebootRequired = $true }
    default {
        Write-Log " -> FAILED, msixec exit code $($proc.ExitCode)"
        $failures++
    }
}

}

if ($failures -gt 0) {
    $exitCode = 1603
} elseif ($rebootRequired) {
    $exitCode = 2
} else {
    $exitCode = 0
}

```

```

}

Write-Log "=== Complete. Exit code: $exitCode ==="
exit $exitCode
}
catch {
    Write-Log "ERROR: $($_.Exception.Message)"
    Write-Log "=== Complete. Exit code: 99 ==="
    exit 99
}

```

Proof
Vulnerable software installed: Microsoft ASP.NET 7.0.20.24269
Vulnerable software installed: Microsoft ASP.NET 7.0.20.24269
Vulnerable software installed: Microsoft ASP.NET 7.0.20.24269
Vulnerable software installed: Microsoft ASP.NET 2.1.30.60071
Vulnerable software installed: Microsoft ASP.NET 7.0.20.24269
Vulnerable software installed: Microsoft ASP.NET 7.0.20.24269
Vulnerable software installed: Microsoft ASP.NET 6.0.36.24516
Vulnerable software installed: Microsoft ASP.NET 7.0.20.24269
Vulnerable software installed: Microsoft ASP.NET 7.0.20.24269
Vulnerable software installed: Microsoft ASP.NET 7.0.20.24269
Vulnerable software installed: Microsoft ASP.NET 7.0.20.24269
Vulnerable software installed: Microsoft ASP.NET 7.0.20.24269
Vulnerable software installed: Microsoft ASP.NET 2.1.30.60071
Vulnerable software installed: Microsoft ASP.NET 7.0.20.24269
Vulnerable software installed: Microsoft ASP.NET 7.0.20.24269

Vulnerable software installed: Microsoft ASP.NET 7.0.20.24269
Vulnerable software installed: Microsoft ASP.NET 7.0.20.24269
Vulnerable software installed: Microsoft ASP.NET 6.0.36.24516
Vulnerable software installed: Microsoft ASP.NET 7.0.20.24269
Vulnerable software installed: Microsoft ASP.NET 7.0.20.24269
Vulnerable software installed: Microsoft ASP.NET 7.0.20.24269
Vulnerable software installed: Microsoft ASP.NET 9.0.9.25420 (
Vulnerable software installed: Microsoft ASP.NET 2.1.30.60071
Vulnerable software installed: Microsoft ASP.NET 7.0.20.24269
Vulnerable software installed: Microsoft ASP.NET 9.0.3.25112 (
Vulnerable software installed: Microsoft ASP.NET 8.0.14.25112
Vulnerable software installed: Microsoft ASP.NET 7.0.20.24269
Vulnerable software installed: Microsoft ASP.NET 7.0.20.24269
Vulnerable software installed: Microsoft ASP.NET 6.0.36.24516
Vulnerable software installed: Microsoft ASP.NET 7.0.20.24269
Vulnerable software installed: Microsoft ASP.NET 7.0.20.24269

Key
HKEY_LOCAL_MACHINE\Software\Wow6432Node\Microsoft\Windows\CurrentVersion\Uninstall\{85662951-18B2-3C85-A696-D58BDBD16A80}
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\{BD401329-F877-391C-9E5A-FEB423C5A196}
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\{BD401329-F877-391C-9E5A-FEB423C5A196}
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\{3F0FA3FE-95FA-3B48-ABD1-46FB4DA4021E}
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\{BD401329-F877-391C-9E5A-FEB423C5A196}
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\{BD401329-F877-391C-9E5A-FEB423C5A196}

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\{0EFFE892-CAC6-338E-997A-8FBFB20B9B1C}
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\{BD401329-F877-391C-9E5A-FEB423C5A196}
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\{BD401329-F877-391C-9E5A-FEB423C5A196}
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\{BD401329-F877-391C-9E5A-FEB423C5A196}
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\{BD401329-F877-391C-9E5A-FEB423C5A196}
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\{BD401329-F877-391C-9E5A-FEB423C5A196}
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\{3F0FA3FE-95FA-3B48-ABD1-46FB4DA4021E}
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\{BD401329-F877-391C-9E5A-FEB423C5A196}
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\{BD401329-F877-391C-9E5A-FEB423C5A196}
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\{BD401329-F877-391C-9E5A-FEB423C5A196}
HKEY_LOCAL_MACHINE\Software\Wow6432Node\Microsoft\Windows\CurrentVersion\Uninstall\{85662951-18B2-3C85-A696-D58BDBD16A80}
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\{0EFFE892-CAC6-338E-997A-8FBFB20B9B1C}
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\{BD401329-F877-391C-9E5A-FEB423C5A196}
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\{BD401329-F877-391C-9E5A-FEB423C5A196}
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\{BD401329-F877-391C-9E5A-FEB423C5A196}
HKEY_LOCAL_MACHINE\Software\Wow6432Node\Microsoft\Windows\CurrentVersion\Uninstall\{9E0B769C-8CB4-3B03-80D4-03D9227DF5DA}
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\{3F0FA3FE-95FA-3B48-ABD1-46FB4DA4021E}
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\{BD401329-F877-391C-9E5A-FEB423C5A196}
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\{60C712CC-7335-3B48-9072-261601C12CDD}
HKEY_LOCAL_MACHINE\Software\Wow6432Node\Microsoft\Windows\CurrentVersion\Uninstall\{49DE7ED6-E9E4-3331-9FBD-E6F16A10FF9C}
HKEY_LOCAL_MACHINE\Software\Wow6432Node\Microsoft\Windows\CurrentVersion\Uninstall\{85662951-18B2-3C85-A696-D58BDBD16A80}

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\{BD401329-F877-391C-9E5A-FEB423C5A196}

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\{0EFFE892-CAC6-338E-997A-8FBFB20B9B1C}

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\{BD401329-F877-391C-9E5A-FEB423C5A196}

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\{BD401329-F877-391C-9E5A-FEB423C5A196}

Revision #3

Created 31 July 2026 17:56:24 by Guest

Updated 31 July 2026 18:44:12 by Christian Kaba