Windows Patch Rollback Worklet

Uninstall a specific Windows patch across multiple devices, Worklet example.

The Automox console allows you to uninstall patches using the Rollback Action from the Device Details page. Unfortunately, this method might be cumbersome for a large number of devices.

So for this scenario, we can apply a Worklet that will allow us to detect the presence of, and subsequently remove, the unwanted patch.

**Note:** Not all patches are uninstallable. Refer to the Microsoft Update Catalog for details for your particular patch.

**Evaluation Code**

To evaluate this, we simply need to determine if the patch is present. This is simple to do with the PowerShell command `Get-HotFix`. Based on the response of that command for the patch we’re concerned with, we use the appropriate Exit Code to indicate its compliance status.

```powershell
#### EVALUATION CODE ####
# If you want to have an ongoing evaluation use this script
# to see if the patch is present.

# If you just want to manually execute this policy, this can be as
# simple as "Exit 1"

# Change this KB number to match what you want to check for
# Be certain to use the same KB number in both Evaluation and Remediation
$kb = '4503308'

# Check for presence and assign to variable
$installed = Get-HotFix -Id "KB$kb" -ErrorAction SilentlyContinue

# Check the variable and exit accordingly
if ( $installed ) {
    # Installed is Non-Conmpliant, so Exit 1
    Exit 1
    # Otherwise Exit 0 for Compliance
} else {
    Exit 0
}
```

**Remediation Code**

**Method 1 - Windows 7 and Newer**

Remediation is more complex in this case. Since Windows 10 removed the option to uninstall patches silently with wusa.exe, we have to dig through packages another way, format the output, and use dism.exe to uninstall the patch.
Method 2 - Windows 8 and Older

For the sake of example, here is the simpler version that can be used on devices using older operating systems (Windows 8 and older). The one complication here lies in the need to use the 'sysnative' path to wusa.exe when running on a 64-bit operating system. So we add a check for that and act accordingly.

Note: This is necessary because Automox runs as a 32-bit process even on 64-bit versions of Windows.

```powershell
#### REMEDIATION CODE ####
# Uninstall the specified patch using wusa.exe
# Compatible with Windows 8 and Older

# Change this KB number to match what you want to check for
# Be certain to use the same KB number in both Evaluation and Remediation
$kb = '4503308'

# Determine OS Architecture to set path for wusa.exe
$osArch = (Get-WmiObject -Class Win32_OperatingSystem).OSArchitecture

# Define the FilePath to wusa.exe based on OS Architecture
if ( $osArch -match '64-bit' ) {
    $filePath = 'C:\Windows\sysnative\wusa.exe'
} else {
    $filePath = 'C:\Windows\System32\wusa.exe'
}

# Uninstall and save the exit code to determine success/failure
Exit $process.ExitCode
```