Using PsExec to Test PowerShell 32bit Scripts on Windows with FileWave

What

FileWave is a Unified Endpoint Management tool that allows organizations to manage and deploy software and settings to macOS, Windows, iOS, iPadOS, tvOS, Chrome, and Android devices. When deploying PowerShell scripts through FileWave, it is important to test the scripts on a Windows device beforehand to ensure that they will function correctly when run through FileWave.

When/Why

FileWave runs all PowerShell scripts in 32bit PowerShell and runs them as SYSTEM. By testing the scripts on a Windows device using PsExec and executing the .ps1 script as SYSTEM with 32bit PowerShell, organizations can gain a better understanding of how the script will function when run through FileWave. This can help prevent potential issues and ensure that the scripts function as intended when deployed to devices.

How

1. Download PsExec from the Sysinternals Suite (https://docs.microsoft.com/en-us/sysinternals/downloads/psexec)
2. Open a Command Prompt or PowerShell window with Administrator privileges
3. Navigate to the folder where PsExec is located
4. Use the following command to run the PowerShell script as SYSTEM with 32bit PowerShell: `pPSEXEC -i -s -d C:\Windows\SysWOW64\windowsPowerShell\v1.0\powershell.exe -file "path\to\script.ps1"`

Example:

```
PSEXEC -i -s -d C:\Windows\SysWOW64\windowsPowerShell\v1.0\powershell.exe -file "C:\test\testscript.ps1"
```

Related Content

- PsExec - Sysinternals | Microsoft Learn
- Script Best Practices

Digging Deeper

PsExec Switches

- `-i` : runs the program in the interactive mode (the user is prompted to confirm the action)
- `-s` : runs the program as the SYSTEM account
- `-d` : runs the program as a background process, without interaction
- `-accepteula` : automatically accepts the PsExec license agreement

Using 64bit PowerShell in a 32bit PowerShell script

Sometimes, certain tasks cannot be accomplished using 32bit PowerShell. To overcome this limitation, the following code can be used to run a 64bit PowerShell script within a 32bit PowerShell script.

```
If ( [IntPtr]::Size * 8 -ne 64 )
{
    C:\Windows\SysNative\WindowsPowerShell\v1.0\PowerShell.exe -File $MyInvocation.MyCommand.Path
}
Else
{
    # Add code here
}
```

This code block checks whether the processor architecture is 64-bit or not and runs the script in 64-bit mode and the main script can be run here. It is important to note that running the script in 64-bit mode should only be done when it is not possible to accomplish a task using 32bit PowerShell.

It is always recommended to test the script in a test environment to avoid any misconfigurations, and to make sure it is working as expected.

Revision #3
Created 2 July 2023 17:18:27 by Josh Levitsky
Updated 4 November 2024 13:43:14 by Sean Holden