Deploy Cisco Secure Endpoint Secure Client Using Microsoft Intune

Contents

Introduction

Background

Configuration

Secure Endpoint Deployment

- Step 1. Download the Cisco Secure Endpoint Installer
- Step 2. Prepare the Intune file using Win32 Content Prep Tool
- Step 3.Upload the Secure Endpoint IntuneWin File to Microsoft Intune Admin Center

Secure Client Deployment

- Step 1. Download the Cisco Secure Client Full Deployment
- Step 2. Prepare the Intune File, Same Procedure as Step 2: This Creates the csc-deploy-full-Intune Test.intunewin File
- Step 3.Upload the csc-deploy-full-intune Test.intunewin File from Stept 1 to Microsoft Intune Admin Center

Install-CiscoSecureEndpoint.ps1 Script

Install-CiscoXDRDeployment.ps1 Script

Confirm-CSEInstall.ps1 Script

Introduction

This document outlines the process for deploying Cisco Secure Endpoint or Secure Client using Microsoft Intune.

Background

This document walks-through the steps in creating Microsoft Intune supported apps from the Secure Endpoint/Secure Client installers and then use for deployment using the Microsoft Intune admin center. Specifically, the process includes packaging the Cisco Secure Endpoint installer as a Win32 application through the Intune Win32 Content Prep Tool, and by configuring and deploying the app through Intune.

Configuration

Secure Endpoint Deployment

Step 1. Download the Cisco Secure Endpoint Installer

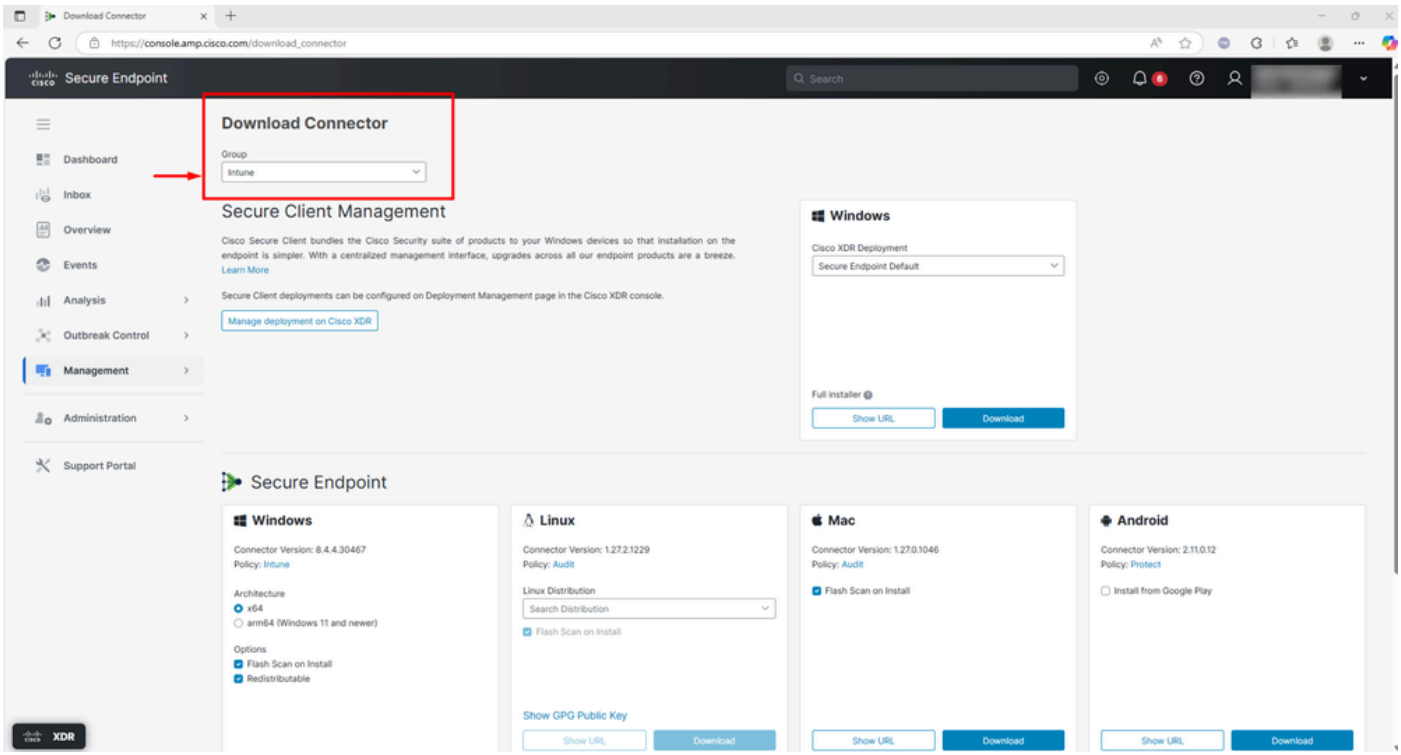
1. Log in to your respective Secure Endpoint Portal, depending on the region.

---

2. Navigate to the **Management** tab and select **Download Connector**.

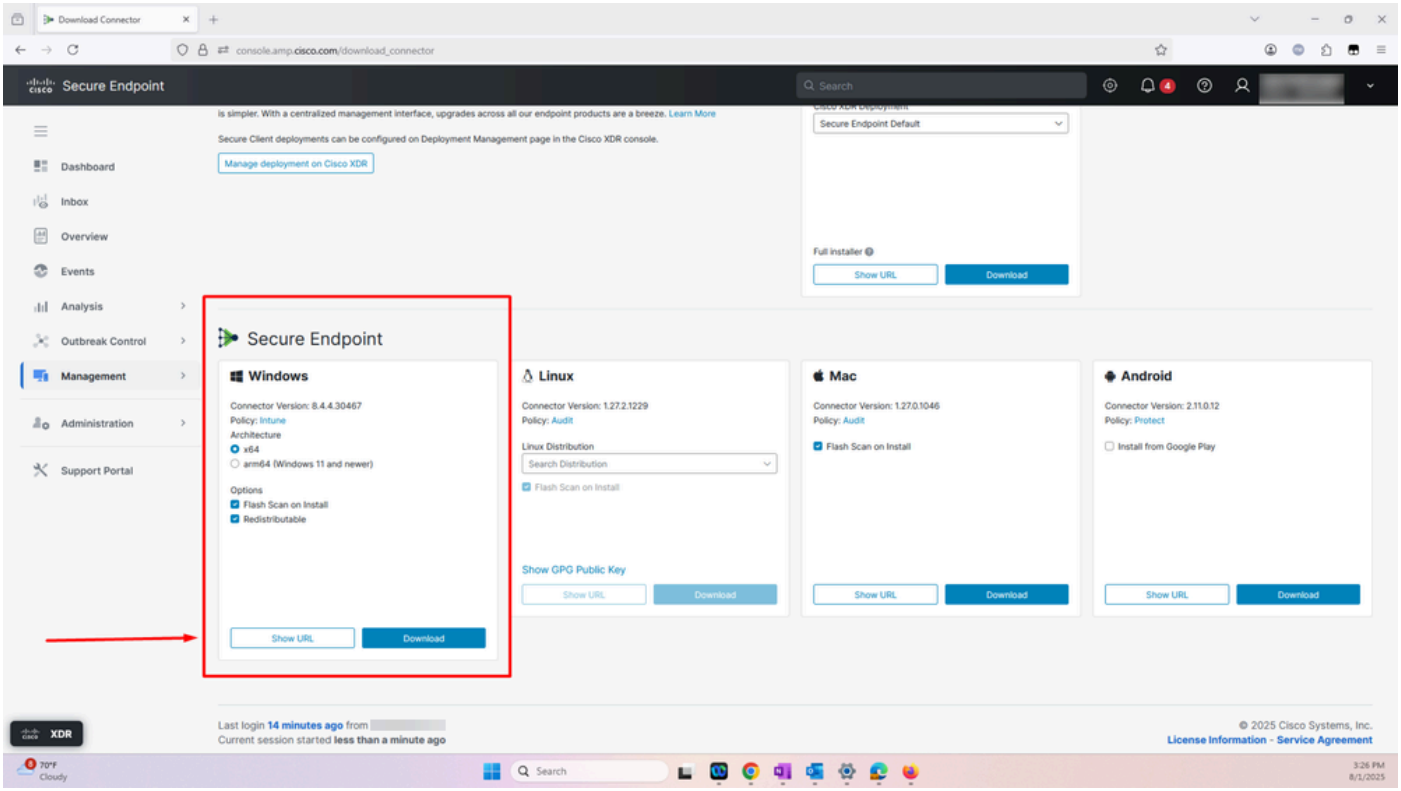
---

3. Select the Secure Endpoint **group** you want the connector to register to.



---

4. Select **download** and the EXE installer is downloaded locally:



Step 2. Prepare the Intune file using Win32 Content Prep Tool

The Win32 Content Prep Tool is a utility provided by Microsoft Intune to help IT administrators prepare Win32 applications (for traditional Windows desktop apps) for deployment through Microsoft Intune. The tool converts Win32 application installers (like .exe, .msi, and related files) into a .intunewin file format, which is required for deploying these apps via Intune.

To prepare the Intune file:

1. Download the Win32 Content Prep Tool from Github.

---

2. Execute **IntuneWinAppUtil.exe**.

---

3. In the next step, change the directory to the folder that contains the Cisco Secure Endpoint executable file that was downloaded in Step 1 and the install powershell script **(Install-CiscoSecureEndpoint.ps1**.)

---

4. Specify the script filename for the setup file, **Install-CiscoSecureEndpoint.ps1**.

---

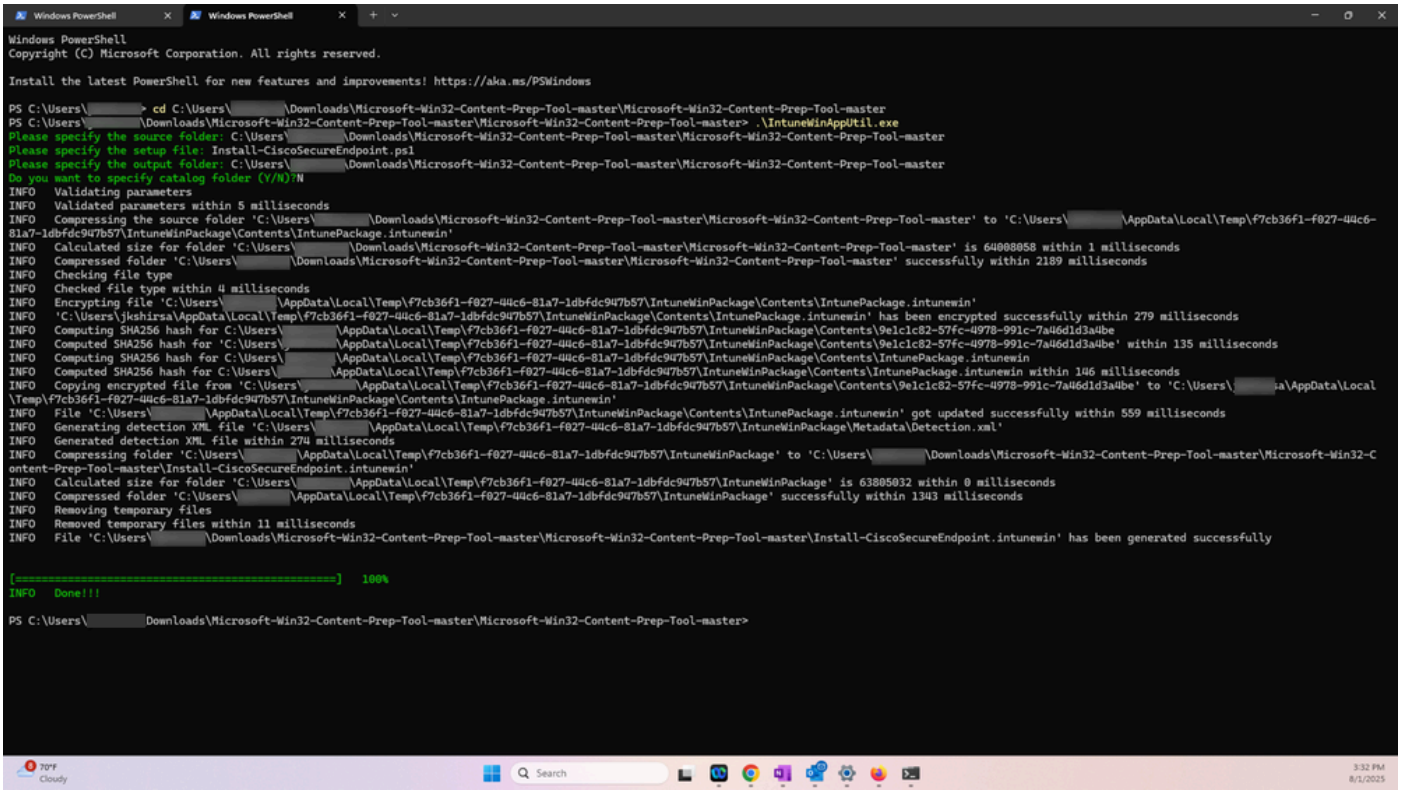
5. In the next step, specify the folder where the **Intunewin** file must be generated.

---

6. Enter **N**, when prompted to specify the catalog.

---

7. The **Intunewin** file is generated as shown in the screenshot:

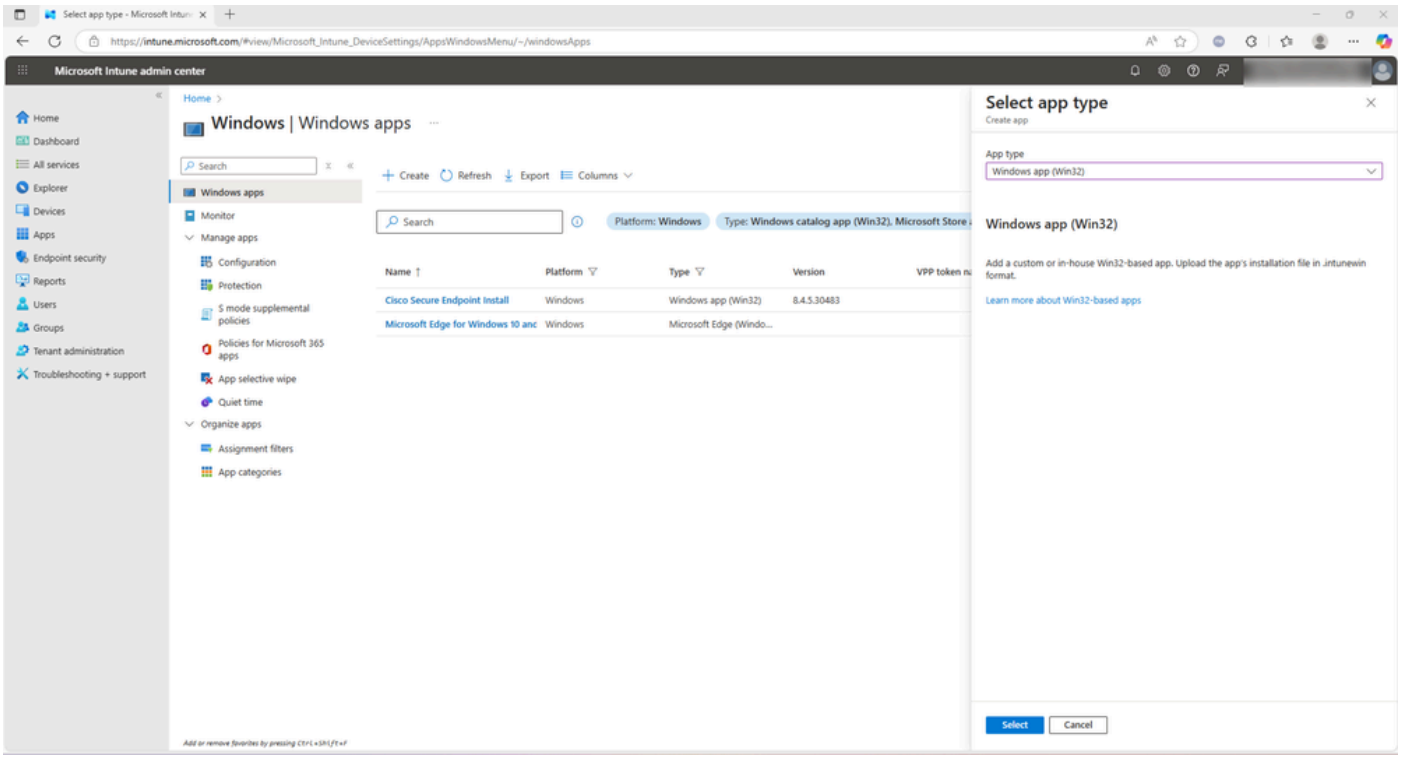


Step 3. Upload the Secure Endpoint IntuneWin File to Microsoft Intune Admin Center

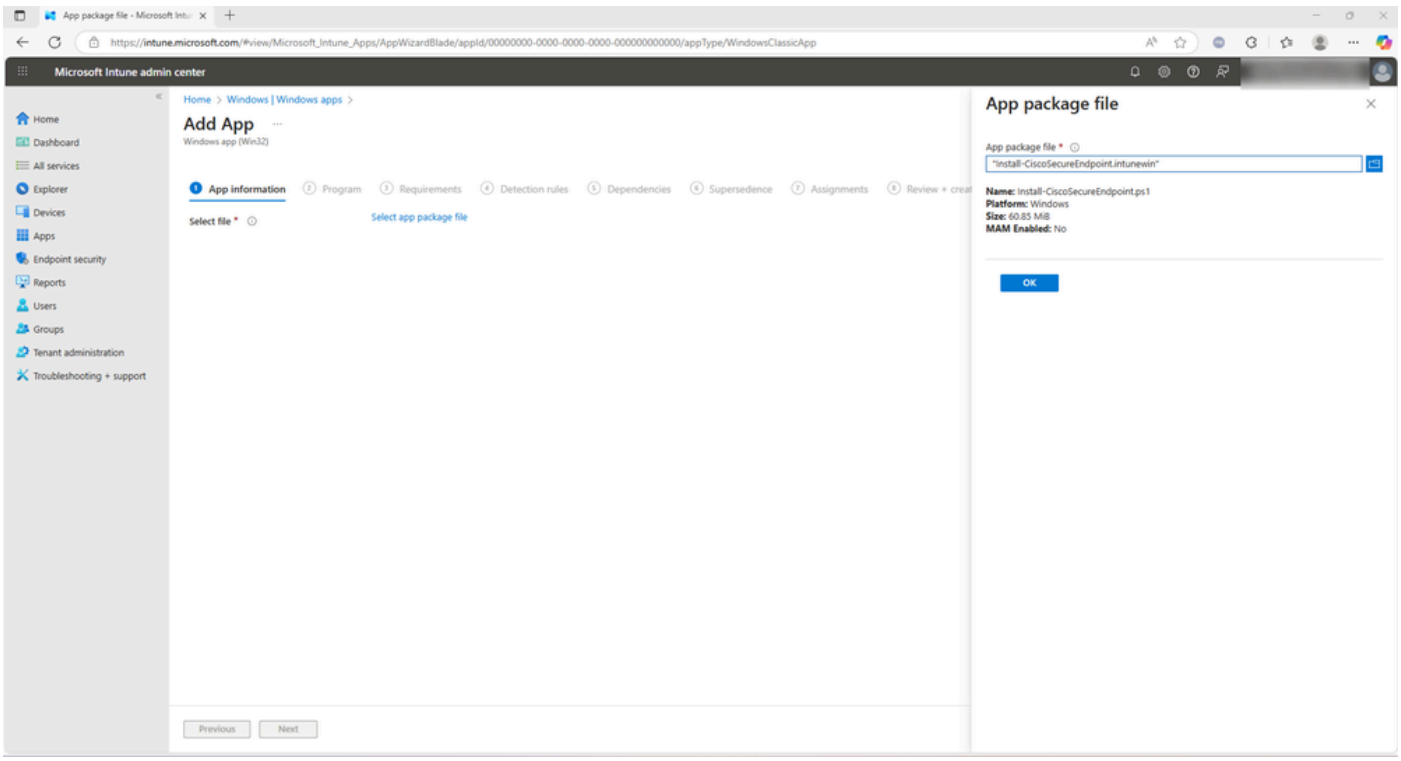
1. Log into **Microsoft Intune Admin Center**.

---

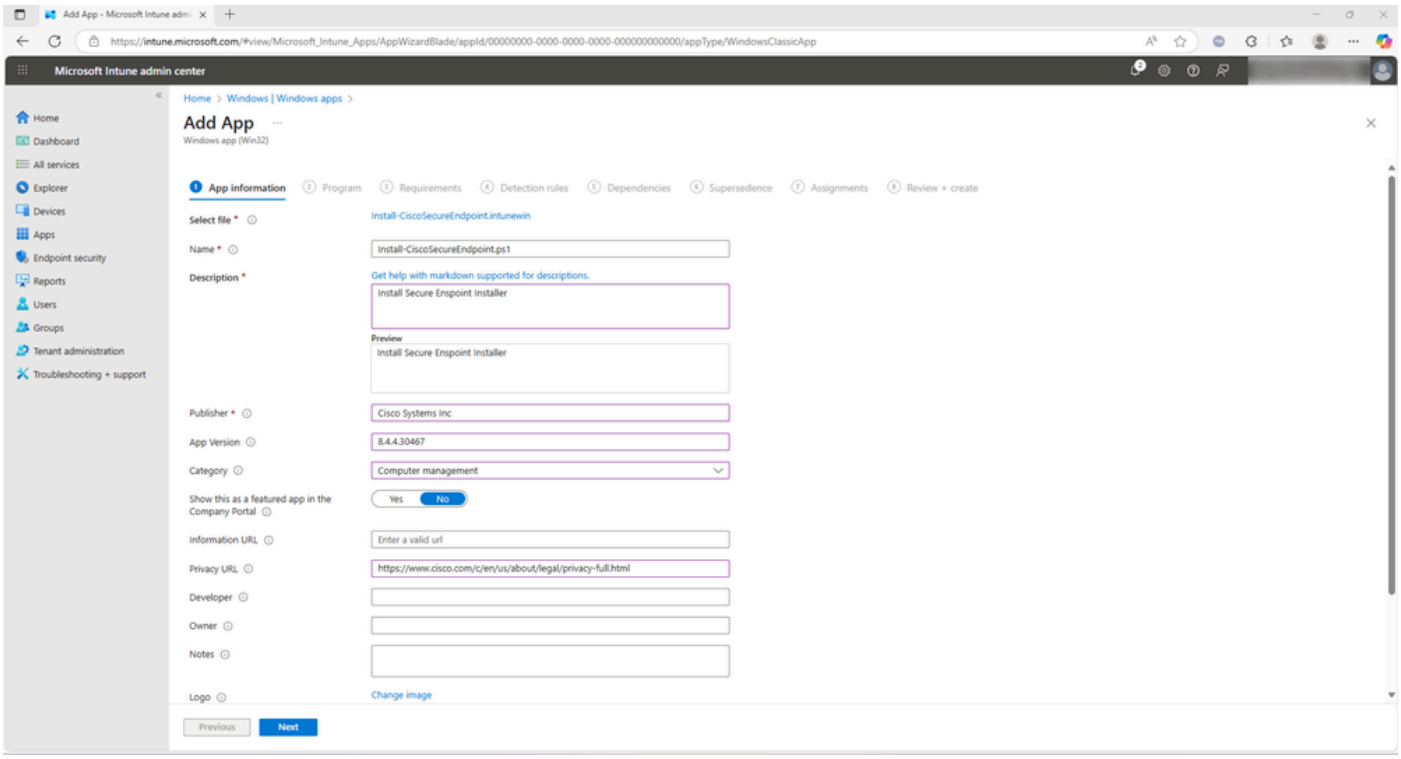
2. Navigate to the **Windows Apps** in the Microsoft Intune Admin Center and Select **App Type – Win32** and select these two actions depicted in the screenshot:



3. In the next step, upload the Secure Endpoint Intunewin file created in Step 2 and select **OK**.



4. After selecting **OK**, enter the information as presented in the screenshot. The optional fields can be left blank on each tab, then click **Next**.



---

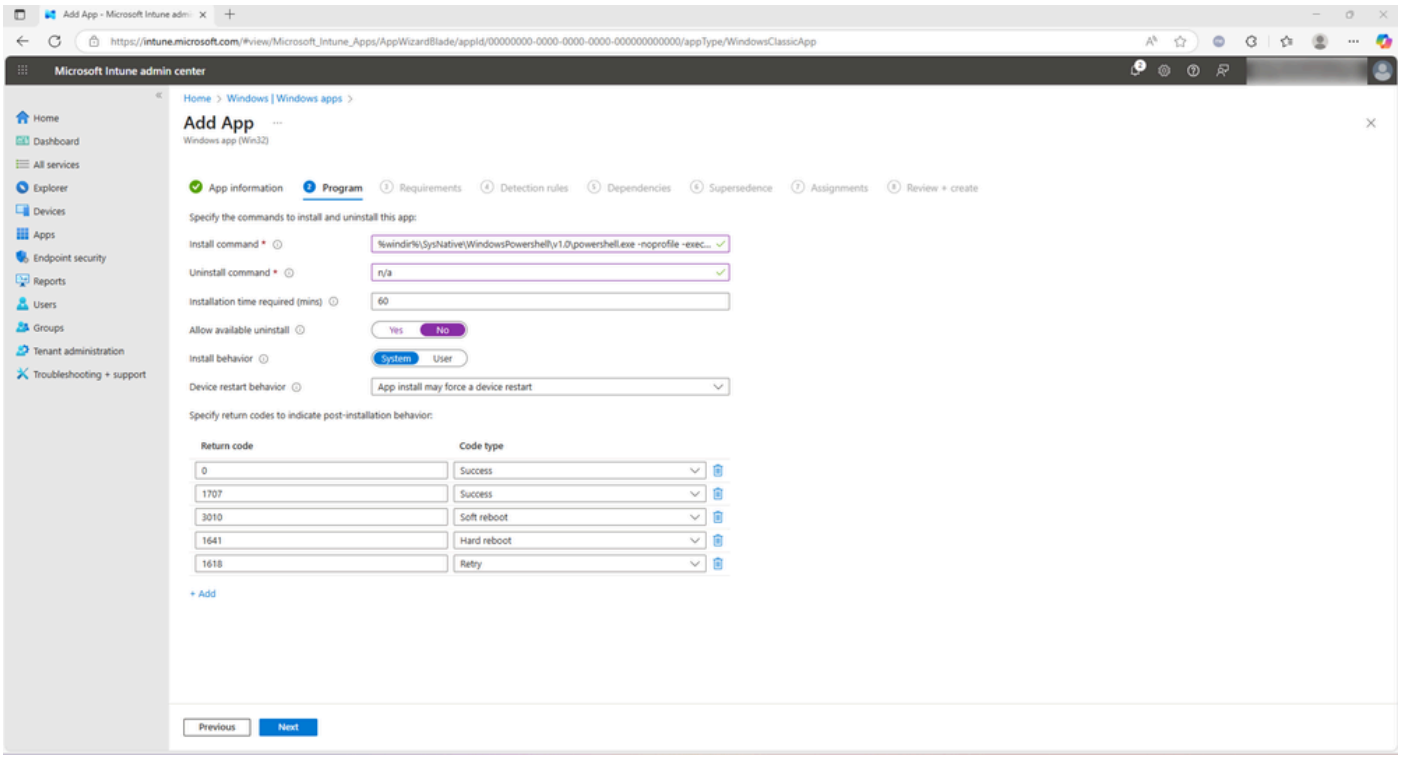
5. Run the **Install** command:

```
%windir%\SysNative\WindowsPowershell\v1.0\powershell.exe -noprofile -executionpolicy Bypass -file .\Ins
```

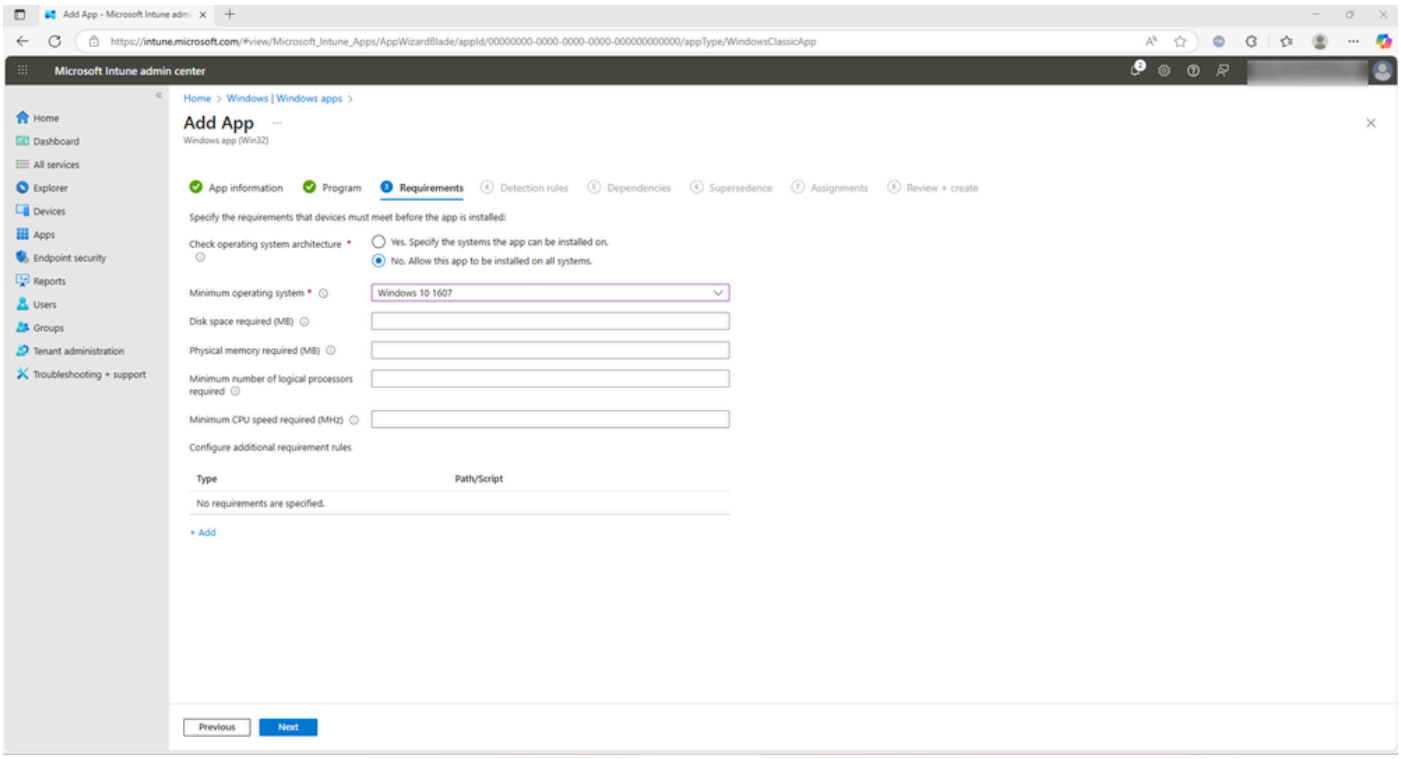
Note: The code presented is an example and any code can be used as an install command for this installer.

---

6. Enter **Uninstall** as n/a and the installation time required as 60 (optional.) Set Allow available uninstall as **No**, and select Install behavior as **System**. Add any additional optional details before you select **Next**.



7. On the requirements tab, check **No. Allow this app to be installed on all systems** and select the minimum operating system. Fill in the optional fields if desired and select **Next**.



8. On the **Detection Rules** tab, the rules format drop-down menu provides two options; **Manually configure detection rules** and **Use a custom detection script**. Either option can be selected based on the

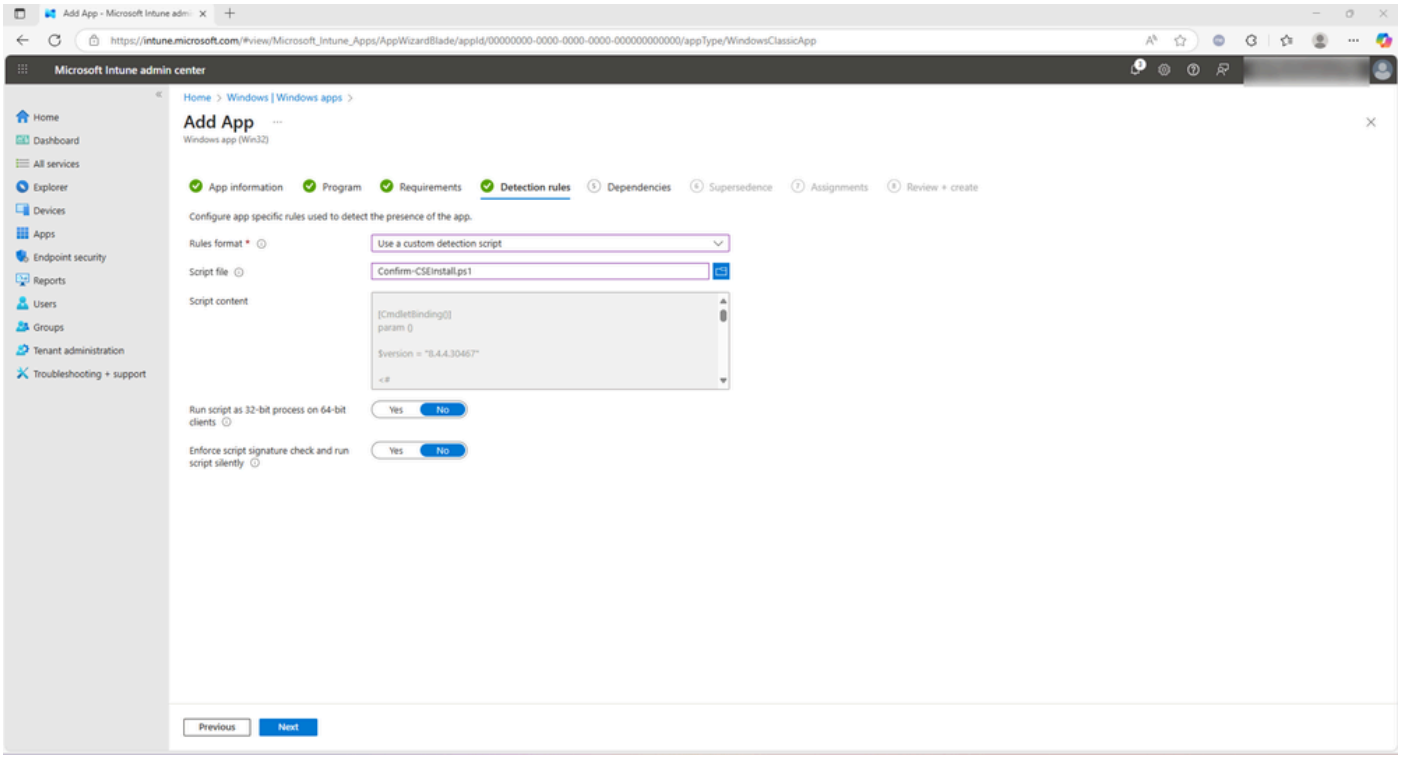
deployment requirements.

---

9. When choosing Manually configure detection rules, you can define a rule type such as MSI, File, or Registry to detect the presence of the application. In this document, the alternative option, **Use a custom detection script**, has been selected.

---

10. A PowerShell script named **Confirm-CSEInstall.ps1** is used to verify successful installation of the Cisco Secure Endpoint (it is listed at the bottom of this document.)



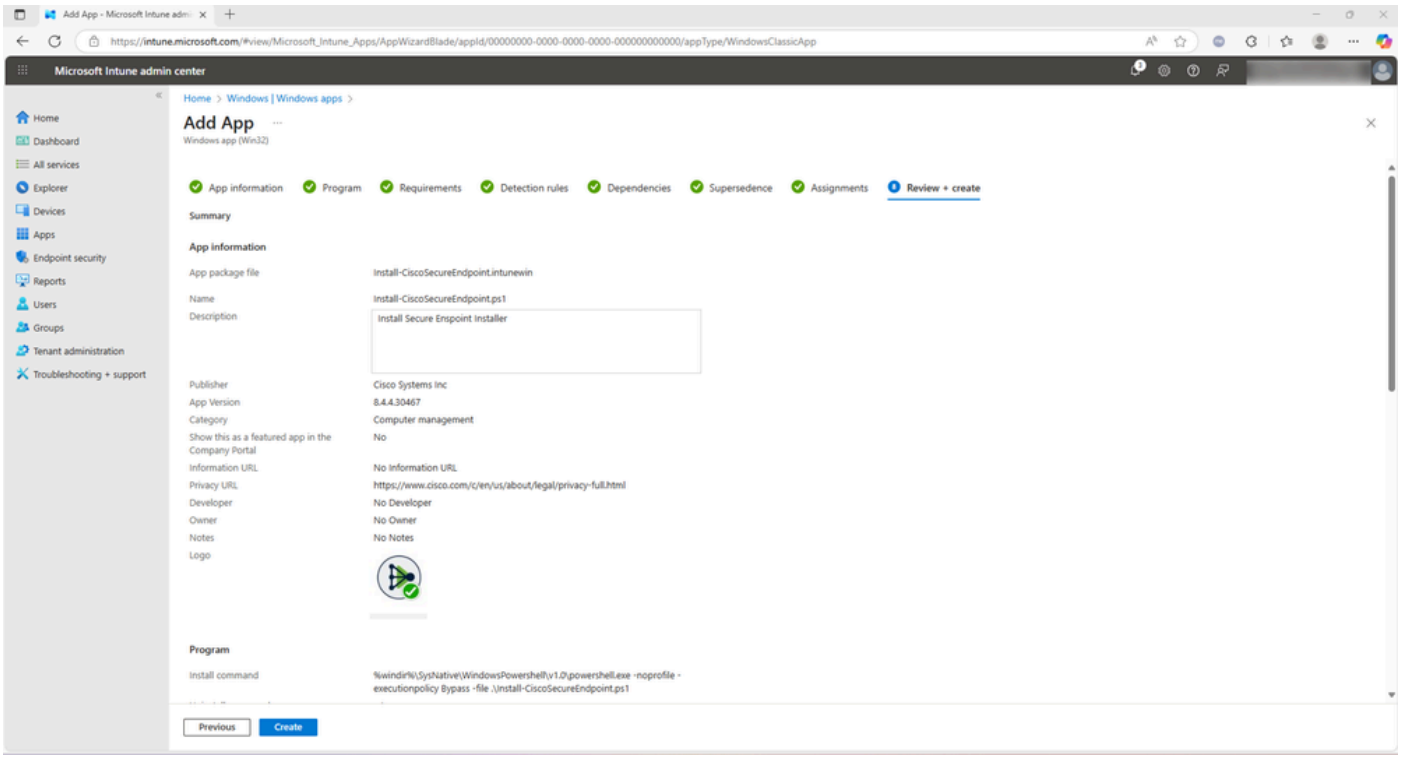
---

11. Select **Next** to proceed.

Note: A custom detection script can be created specifically for this deployment process to suit your environment and detection criteria.

---

12. (The next few tabs are optional) No dependencies must be configured; assign the application to the required group and select **Review + create**.



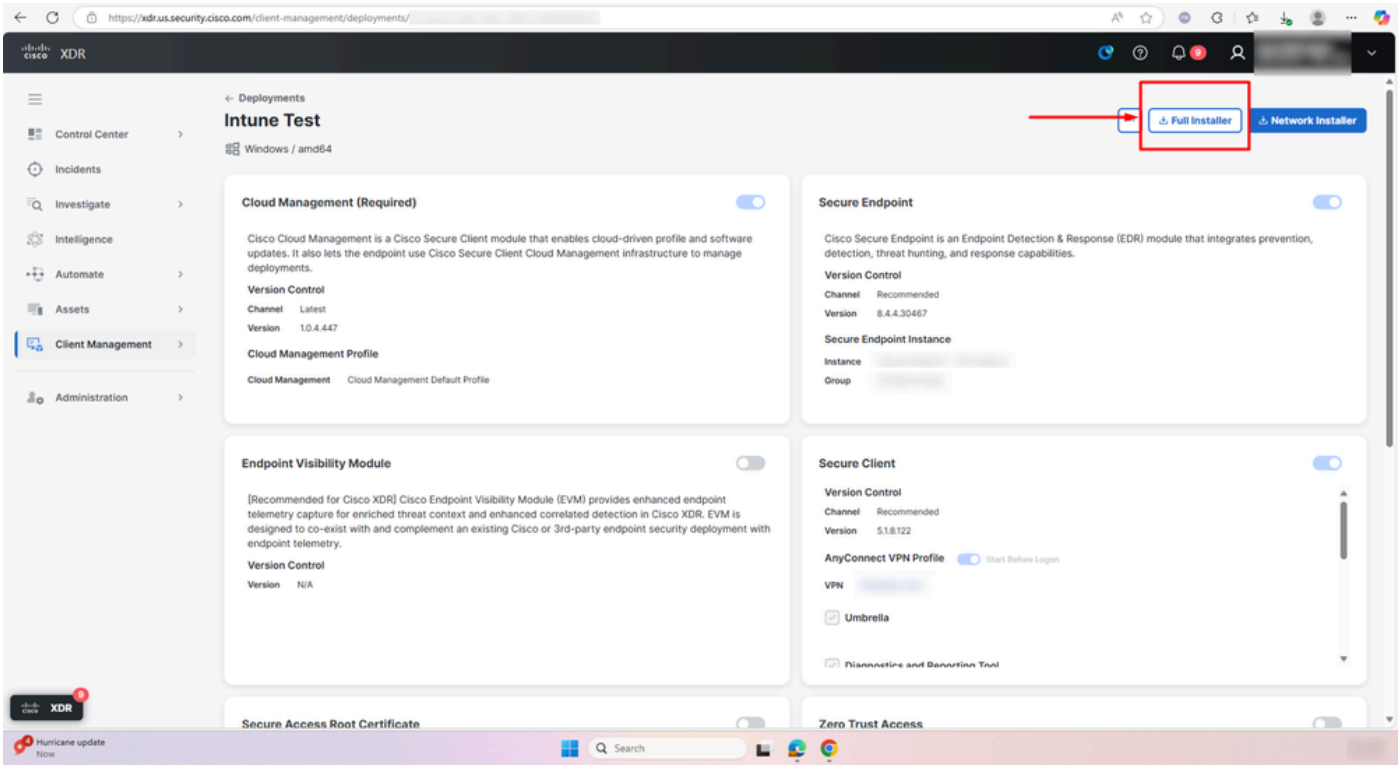
Secure Client Deployment

Step 1. Download the Cisco Secure Client Full Deployment

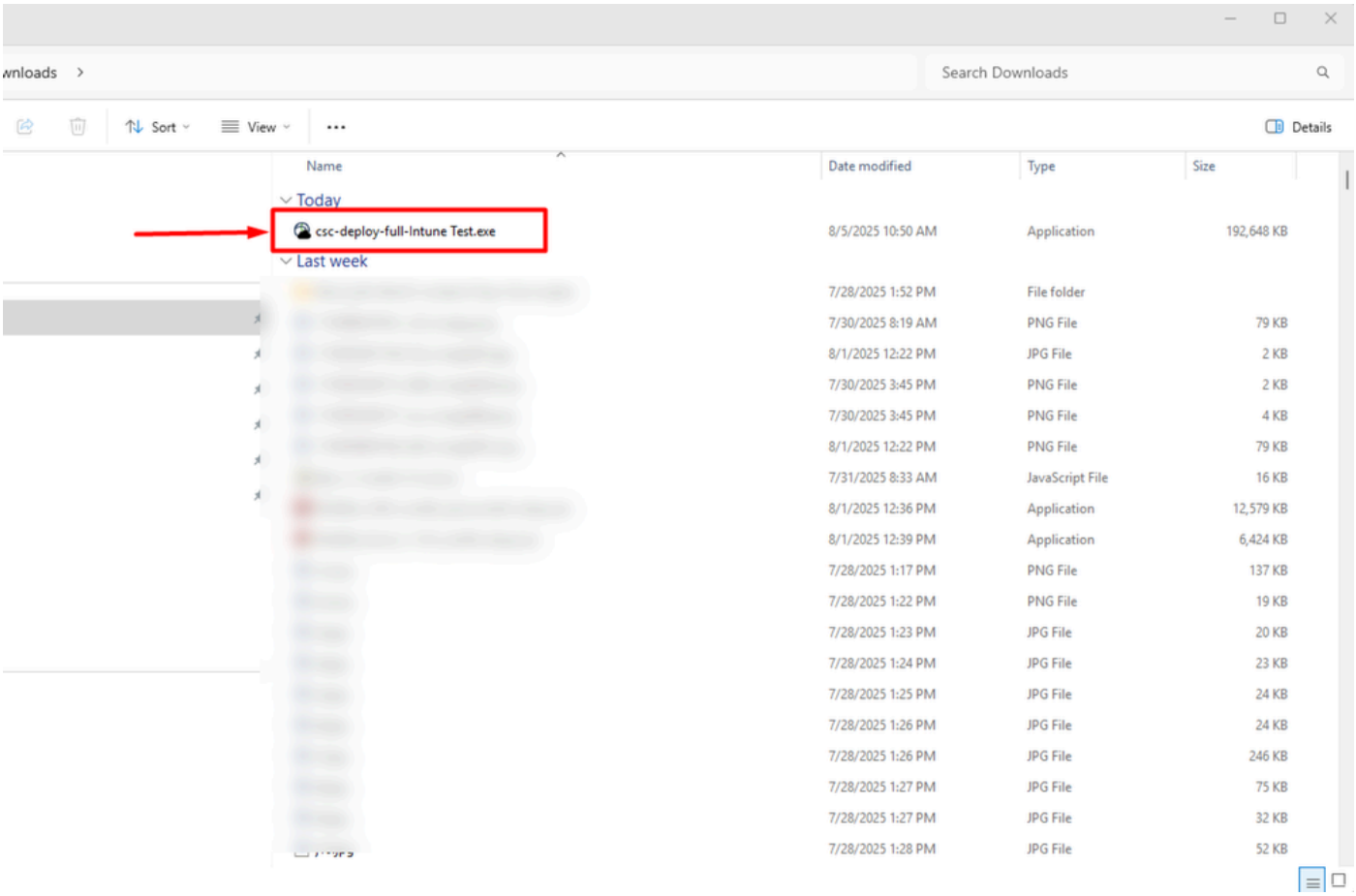
1. Log in to the XDR or Secure Client Cloud Management console, depending on the region.

---

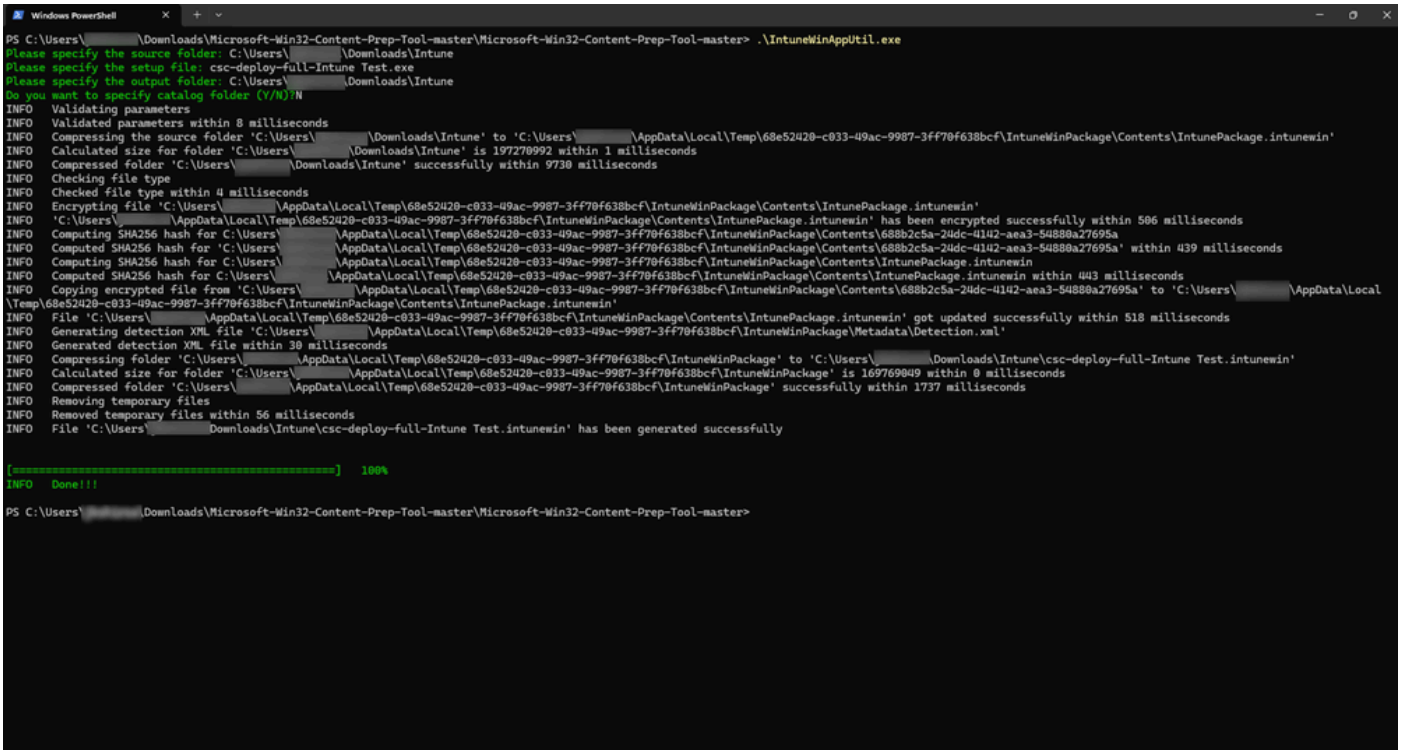
2. Create a new deployment and select **Full Installer** or **Network Installer** depending on your deployment type.



3. A **csc-deploy-full-Intune Test.exe** is downloaded as shown in the screenshot:

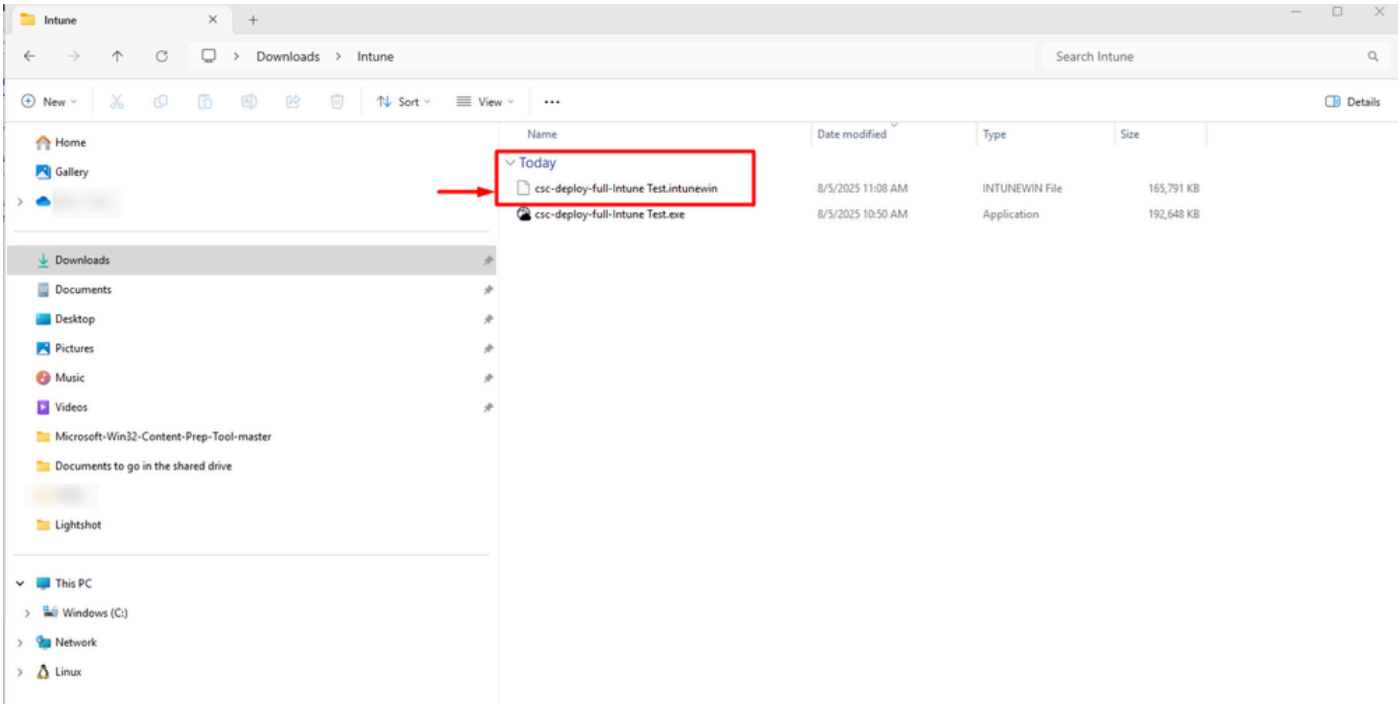


Step 2. Prepare the Intune File, Same Procedure as Step 2: This Creates the csc-deploy-full-Intune Test.intunewin File



---

1. The previous screenshot steps results in creating a csc-deploy-full-Intune Test.intunewin file.



Step 3. Upload the csc-deploy-full-intune Test.intunewin File from Stept 1 to Microsoft Intune Admin Center

This completes the process to deploy Cisco Secure Endpoint using Intune.



Caution: This install script must be combined with the installer using Intune Converter and the exact name of the PS script referenced in the command line mentioned under Step 3 in this guide and placed in the Intune under:

Add App > Program > Install Command.



Note: Scripts in the next section are working samples for demonstrations used in lab environments to create this guide. You can use them as-is after proper testing or modify as you wish.

Install-CiscoSecureEndpoint.ps1 Script

```
[CmdletBinding()]
param ()

$cse_exe = <Secure Endpoint Installer>
$version = <Secure Endpoint Version>

if ($PSCommandPath -eq $null) {
    function GetPSCommandPath() {
        return $MyInvocation.PSCommandPath;
    }
    $PSCommandPath = GetPSCommandPath
}

$script = [pscustomobject]@{
    "Path" = Split-Path $PSCommandPath -Parent
    "Name" = Split-Path $PSCommandPath -Leaf
}

Set-Location -Path $script.Path

$cse_installer = [IO.Path]::Combine($script.Path, $cse_exe)
$csc_installer_args = "/R /S"

<#
    Cannot use -wait for 'Cisco Secure Endpoint' and therefore cannot get the exit code to return.
    Using -wait, returns varied results, instead use Get-Process and while loop to wait for installatio
#>
$install = Start-Process -WorkingDirectory "$($script.Path)" -FilePath "${cse_installer}" -ArgumentList
```

```
while (Get-Process "$($cse_exe -replace '.exe', '')" -ErrorAction SilentlyContinue)
{
    Start-Sleep -Seconds 10
}
```

Install-CiscoXDRDeployment.ps1 Script

```
$exeName = <Your_XDR_Deployment.exe>
$exePath = Join-Path -Path $PSScriptRoot -ChildPath $exeName

if (-not (Test-Path $exePath)) {
    Write-Error "Executable not found: $exePath"
    exit 1
}

# Use silent mode - the most commonly documented switch for Cisco XDR / Secure Client installers
$arguments = "-q"   # or try "--quiet" if -q fails in your specific build

try {
    $process = Start-Process -FilePath $exePath -ArgumentList $arguments -NoNewWindow -Wait -PassThru

    if ($process.ExitCode -eq 0) {
        Write-Output "Cisco XDR installation completed successfully (exit code 0)."
        exit 0
    } else {
        Write-Error "Installation failed with exit code: $($process.ExitCode)"
        exit $process.ExitCode
    }
}
catch {
    Write-Error "Exception during installation: $_"
    exit 1
}
```

Confirm-CSEInstall.ps1 Script



Tip: Optional: You can use the Manually Configured Detection rules in the Intune Management Portal.

```
[CmdletBinding()]
param ()
```

```
$version = <Secure Endpoint Version>

<#
    https://learn.microsoft.com/en-us/intune/intune-service/apps/apps-win32-add#step-4-detection-rules
    The app gets detected when the script both returns a 0 value exit code and writes a string value to

    The Intune agent checks the results from the script. It reads the values written by the script to t
    the standard error (STDERR) stream, and the exit code. If the script exits with a nonzero value, th
    the application detection status isn't installed. If the exit code is zero and STDOUT has data, the
    detection status is installed.
#>

$cse = Get-ItemProperty HKLM:\Software\Microsoft\Windows\CurrentVersion\Uninstall\*, HKLM:\SOFTWARE\Wow
if ($cse | Where-Object { [System.Version] $_.DisplayVersion -ge [System.Version] "${version}" })
{
    Write-Host "Installed"
    exit 0
}

exit 1
```



Note: The previous scripts are only for educational purposes and not supported by Cisco TAC. Each environment can vary with different rules or policies and it is necessary to conduct proper testing. Troubleshooting third party deployment tools is treated as best effort and must start with reviewing Intune logs both in the portal or on the managed station.
