How to Deploy the Barracuda Content Shield Suite With Microsoft Intune

https://campus.barracuda.com/doc/96768320/

Azure connected devices can use Microsoft Intune to remotely manage connected endpoints. This article outlines using Intune to push the BCS Plus agent to a connected endpoint. The administrator creates an .intunewin Win32 App package (a compressed folder with all files necessary for deployment), which is then pushed to the endpoint to be executed with the *install* action. Additionally, an *uninstall* action can be configured using the CLI *uninstall* command. For forced removal of failed uninstall actions, you can run a powershell script to remove the BCS agent and clean up the machine.

Step 1. Create a deployment package

First, create an '.intunewin' package which is a compressed folder with all files necessary for the deployment, and which can be pushed to the endpoint when it is configured to be executed. You can download a tool to create this package. For more information on this process, see ttps://docs.microsoft.com/en-us/mem/intune/apps/apps-win32-prepare .

To create the package:

1. Download the Win32 Content Prep Tool.
2. Log into BCS:
   - Download `BarracudaContentShieldSetup-<VERSION>.exe`
   - Download the account configuration file, `bcs.key`
3. In this example, you can use the sample configuration script, ConfigureBCS.ps1, that controls the installation process and creates logging for later review.
4. Put `BarracudaContentShieldSetup-<VERSION>.exe`, `bcs.key` and `ConfigureBCS.ps1` into the same folder.
5. Place the Win32 Content Prep Tool outside the folder, so that the folder structure looks something like this:
   - IntuneWinAppUtil.exe
   - Barracuda
     - `BarracudaContentShieldSetup-1.2.3.4.exe`
     - `bcs.key`
     - `ConfigureBCS.ps1`
6. Run `IntuneWinApputil.exe` and fill in the data according to instructions in the link above.
   - Pass "`BarracudaContentShieldSetup-1.2.3.4.exe`" as the setup file name. This action names the package.
   - When asked if you want to create a catalog file, select 'N' for No.
7. The tool then creates the `.intunewin` package containing the 3 files listed above.

Step 2. Configure and push a Win32 App via Intune to connected Windows device

1. Use this documentation to add a **Win32 App** to the catalog:
   - https://docs.microsoft.com/en-us/mem/intune/apps/apps-add
   - https://docs.microsoft.com/en-us/mem/intune/apps/apps-win32-add
2. This brings up a configuration template to fill in.
3. Upload the `.intunewin` package to the template and fill in the application information: Name, Description, Publisher, App Version, etc.
4. Fill in the Install command and the Uninstall command. This part uses the Barracuda customized powershell script, which comes bundled with the `.intunewin` package. You just need to enter the CLI command for install ('Install Command') in the appropriate sections. Using the example script `ConfigureBCS.ps1`, the commands for full installation are as follows:
   - **Install Command**: `powershell -ExecutionPolicy Bypass -File ConfigureBCS.ps1 -action install -workDir C:\Barracuda -setupName BarracudaContentShieldSetup-1.2.3.4.exe -keyName bcs.key`
   - **Uninstall Command**: This example uses the uninstall CLI command for the installer instead of the *remove* action of the `ConfigureBCS.ps1` script. For details see Uninstalling the Suite below.
5. Make sure to execute the commands in System context (default setting).
6. For the Detection Rules, check that the registry key for the suite installation exists: `HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\Microsoft\Windows\CurrentVersion\Uninstall\{3877F3A3-358E-49C2-BFA7-9A63994D2FCA}`
7. Since the `setup.exe` takes care of the prerequisites, there are no further dependencies to be configured.
8. Finally, under **Assignments**, assign this app to *All Devices* (since there is only one device in the example). You should have configured a test group that contains test devices, so that you can assign this app to that group. Schedule this as as soon as possible, and make sure to show a notification when the installation is going to start on the target device, so you know when it gets installed.
9. Continue to the Review and Create step. If you're happy with the setup, click **Create**.
10. The endpoint device checks in every hour and will retry a failed action by default 3 times before giving up. The device checks in after a reboot; it takes about 3 to 5 mins before doing so. Once the device receives the new policy, it will display the toast notification and then download the `.intunewin` package from network / server, which may take a few minutes. The package is installed in the background, and silently. The result of the action is reported on Intune Dashboard.

Uninstalling the suite

Win32 App config: Use default CLI command for Uninstall

To uninstall, run the CLI command:

```
BarracudaContentShieldSetup-{version}.exe USER_PASS="my_user_pass" /remove
/silent
```

However, if the Tamper Proof feature is enabled and a password is set, the uninstall will fail if the password has not yet been synced to the endpoint. The same is possible if Tamper Proof was just disabled, but the endpoint is not up to date with the sync. If the uninstall fails, you can run the `ConfigureBCS.ps1` script for forced uninstall and cleanup on the device.

Prepare Cleanup script: A forced uninstall script pushed on demand to select devices that need cleanup

Create a copy of `ConfigureBCS.ps1` and preconfigure the script to be run without extra input parameters. Modify the parameter section in the script header by doing the following:

- Remove the *Mandatory* flag from the first two parameters
- Set the action to *Remove* and the workDir to the directory of your choice.
- Configure logEventLog to be *1* if you want to run this script with redirecting logging to the application event log (see the parameter description here).

```
param(

    [Parameter(HelpMessage='Specify "install" or "remove"')]
    [string]$action='remove',
    [Parameter(HelpMessage='Specify working directory. Gets created if does
not exist and setup files are copied here.')]
    [string]$workDir='C:\MyBarracuda',

    [Parameter(HelpMessage='Required for "install" action only. BCS setup
executable file name')]
    [string]$setupName='',

    [Parameter(HelpMessage='Required for "install" action only. BCS account
configuration (*.key) file name')]
    [string]$keyName='',

    [Parameter(HelpMessage='Required for customized "install" only. Set to
"WebFiltering" or "MalwarePrevention". Not setting this defaults to install
both.')]
    [string]$feature='',
    [Parameter(HelpMessage='Optional. Pass version, i.e. "2.0.0.0", if original installer filename is renamed.')]
    [string]$version='',

    [Parameter(HelpMessage='Optional. Pass this flag and set to "1" if logging to Event log Application.evtx')]
    [string]$logEventLog='1',

    [Parameter(HelpMessage='User password for uninstall (obsolete, since tamperproof gets disabled in this script)')]
    [string]$userPass=''

) #Must be the first statement in the script
```

- Save this script as `RemoveBCS.ps1` and upload it to the Scripts section. Configure the script to run as a Powershell script on a 64bit host in System Context. Make sure signature check is disabled if you do not sign your script so that it will run properly.
- Assign your device to run this script.
- Reboot your device and wait for a few minutes until the script is executed by the Intune ExtensionManager. You can check the workDir for the updated install.log or the Application Event Logs for an update, depending on whether you set your script to log to the Events log or not.
- Check the logs for the result of the script.

When you use the **Scripts** section to run the Remove action, make sure you remove the device assignment after you have uninstalled the applications from your device.