

VMware Carbon Black Cloud Sensor Installation Guide

13 October 2022

VMware Carbon Black Cloud

You can find the most up-to-date technical documentation on the VMware website at:

<https://docs.vmware.com/>

VMware, Inc.
3401 Hillview Ave.
Palo Alto, CA 94304
www.vmware.com

Copyright © 2017-2022 VMware, Inc. All rights reserved. [Copyright and trademark information.](#)

Contents

Preface 8

1 Getting Started with Sensor Installation 9

Before you Install Sensors on Endpoints 9

About Sensor Groups and Policy Assignments 10

Local Scan Settings 10

Setting Antivirus Exclusion Rules 10

Method 1: Invite Users to Install Sensors on Endpoints 12

Invite Users to install Sensors 12

Send a new Installation Code 13

Method 2: Install the Sensor on the Endpoint by using the Command Line or Software Distribution Tools 13

Obtain a Company Registration Code 14

Download Sensor Kits 15

Installing Sensors on Endpoints 16

2 Installing Linux Sensors on Endpoints 17

Unpack the Agent 18

Verify the Unpacked Tar-ball Contents 18

Prerequisites for Linux 4.4+ Kernels for Linux Sensor Versions 2.10+ 19

About the Linux Sensor cfg.ini File 21

Linux Installer Command Line Parameters 23

Install a Linux Sensor on a Single Endpoint 24

Install a Linux Sensor on an Endpoint using the RPM/DPKG Installer 24

Install a Linux Sensor on an Endpoint that Automatically Registers the First Time it is Started 25

3 Installing macOS Sensors on Endpoints 26

Approve the Kernel Extension (macOS 10.13 – macOS 11) 27

Manually Approve the KEXT 27

Approve the KEXT via MDM 27

Approving the KEXT via MDM for Big Sur 28

Identify Devices with Sensors that do not support the Operating System or need KEXT Approval 28

Approving the System Extension and Network Extension for macOS 11+ 29

Approve the System Extension via MDM 29

Approve the Network Extension Component of the System Extension via MDM 30

Full Disk Access Requirement for the macOS Sensor 31

Manually Grant the pre-3.5.1 Sensor Full Disk Access 32

| | |
|-----------------------------------------------------------------------------------|----|
| Manually Grant the 3.5.1 or Later Sensor Full Disk Access | 32 |
| Grant the 3.51+ Sensor Full Disk Access via MDM | 35 |
| Restart Requirements for macOS 10.15+ | 37 |
| Special Considerations for the macOS Sensor on Big Sur | 37 |
| Install a KEXT-enabled Sensor on Big Sur | 37 |
| Switch the macOS Sensor Kernel Type on Big Sur during Update | 38 |
| Toggle between Kernel Extension and System Extension in Big Sur | 38 |
| Manually Install and Approve the Sensor on macOS 11+ | 39 |
| macOS Sensor Command Line Install | 42 |
| Extract and Prepare the macOS Install Files | 42 |
| Perform a macOS Sensor Command Line Installation | 43 |
| macOS Command Line Parameters | 43 |
| macOS Command Line Install Examples | 45 |
| Deploying macOS Sensors on Big Sur and Later by using Jamf Pro | 45 |
| Obtain and Prepare the Sensor | 46 |
| Create a Package by using Jamf Composer | 46 |
| Modify the Installation Script | 48 |
| Upload macOS Sensor DMG and Installation Script to Jamf Pro | 48 |
| Creating a Configuration Profile | 49 |
| Configure Configuration Profile General Settings | 49 |
| Set Privacy Preferences Policy Control in the Configuration Profile | 49 |
| Enable System Extension Payloads in the Configuration Profile | 50 |
| Enable Kernel Extension Payloads in the Configuration Profile | 51 |
| Set the Content Filter in the Configuration Profile | 52 |
| Create a Software Distribution Policy | 53 |
| Create and Assign Smart Computer Groups | 54 |
| Validate a Healthy System Extension Sensor through RepCLI | 55 |
| Address the Extension Warning Post-install | 56 |
| macOS Services, Utilities, and Uninstaller | 57 |
| Installing macOS Sensors on Endpoints by using Workspace ONE UEM | 58 |
| Prepare to Install macOS Sensors | 58 |
| Deploying the Carbon Black Cloud sensor for macOS Manually with System Extensions | 59 |
| Creating a Configuration Profile | 60 |
| Manually Install and Approve the Sensor on macOS 11+ | 65 |
| Confirm the Carbon Black Cloud Sensor Installed on the macOS Device | 67 |
| Deploying the Carbon Black Cloud sensor for macOS as Managed Application | 68 |
| Set Up the Application Installer | 69 |
| Install the Carbon Black Cloud Sensor for macOS as a Managed Application | 70 |
| Confirm the Carbon Black Cloud Sensor for macOS Installed as Managed Application | 74 |

4 Installing Sensors on Endpoints in a VDI Environment 76

| | |
|-------------------------------------------------------------------------------------------------|-----|
| Creating Multiple Golden or Primary Images | 76 |
| Carbon Black Windows Sensors with VMware Horizon Virtual Desktops | 77 |
| Carbon Black Windows Sensor Policy Setting Recommendations for Horizon Instant Clones | 77 |
| Install the Carbon Black Windows Sensor on Horizon Instant Clones | 80 |
| Horizon Instant Clones and Carbon Black Windows Sensor Installation Known Issues and Mitigation | 83 |
| Carbon Black Windows Sensor Policy Setting Recommendations for Horizon Full Clones | 84 |
| Install Carbon Black Windows Sensors on Horizon Full Clones | 86 |
| Install Carbon Black Windows Sensors in Horizon Full and Instant Clone Mixed Environments | 88 |
| Horizon Linked-Clones and Carbon Black 3.6+ Windows Sensor Best Practices | 89 |
| Carbon Black Linux Sensors with VMware Horizon Virtual Desktops | 90 |
| Carbon Black Linux Sensor Policy Setting Recommendations for Horizon Golden Images | 90 |
| Horizon Golden Image Considerations for Carbon Black Linux Sensors | 91 |
| Horizon Instant Clone Considerations for Carbon Black Linux Sensors | 91 |
| Install the Carbon Black Linux Sensor on a Horizon Golden Image and Create Instant Clones | 91 |
| Carbon Black Windows Sensors with Citrix Virtual Desktops | 93 |
| Citrix Golden Image Considerations for Carbon Black Sensors | 93 |
| Carbon Black Policy Setting Recommendations for Citrix Golden Images | 94 |
| Install the Sensor on a Citrix Golden Image | 95 |
| Citrix Clone Considerations for Carbon Black Windows Sensors | 97 |
| Carbon Black Policy Setting Recommendations for Citrix Clones | 97 |
| Citrix MCS and Carbon Black Windows 3.7MR1+ Sensor | 99 |
| Citrix PVS and Carbon Black Windows 3.7MR1+ Sensor | 100 |
| Citrix PVS and Carbon Black Windows 3.7 Sensor | 101 |
| Carbon Black Windows Sensors with vSphere Clients | 102 |
| Carbon Black Linux Sensors with vSphere Clients | 104 |

5 Installing Windows Sensors on Endpoints 106

| | |
|-------------------------------------------------------------------------|-----|
| Verifying Windows Sensor Digital Signatures | 107 |
| Windows Sensor Rollback | 110 |
| Local Scan Settings and the AV Signature Pack | 112 |
| To Disable Automatic Signature Updates and use the Standalone Installer | 112 |
| To Update the AV Signature Pack by using the RepCLI Command | 113 |
| Windows Sensor Command Line Parameters | 113 |
| Windows Sensor Supported Commands | 114 |
| Obfuscation of Command Line Inputs | 118 |
| Windows Command Line Install on Endpoints — Examples | 118 |
| Windows Sensor Log Files and Installed Services | 119 |

| | |
|-----------------------------------------------------------------------------------------------------------|-----|
| Installing Windows Sensors on Endpoints by using Group Policy | 120 |
| Create a Microsoft Installer Transform (.MST) File | 120 |
| Automatically Create a Windows Installer .MSI Log | 121 |
| Install Sensors by using Group Policy | 121 |
| Installing Windows Sensors on Endpoints by using SCCM | 122 |
| Add the Sensor Application to SCCM | 122 |
| Deploy the Sensor Application using SCCM | 124 |
| Verify that the Sensor Application was Deployed via SCCM | 125 |
| Installing Carbon Black Cloud Sensor for Windows by Using Workspace ONE UEM | 126 |
| Deploy the Carbon Black Cloud Sensor for Windows as Managed Application in Workspace ONE UEM | 126 |
| Verify that Carbon Black Cloud Sensor for Windows Installed as Managed Application with Workspace ONE UEM | 127 |

6 Search for Sensors 128

7 Updating Sensors on Endpoints 129

| | |
|--------------------------------------------------------------|-----|
| About Updating Sensors on Endpoints through the Console | 130 |
| Update Sensors on Endpoints through the Console | 130 |
| Update Windows Sensors on Endpoints through the Command Line | 132 |
| Update Sensors on Endpoints by using Group Policy | 132 |
| Update Sensors on Endpoints that were Deployed by using SCCM | 133 |
| Update Linux Sensors on Endpoints through the Command Line | 134 |
| View Progress of Sensor Updates | 135 |
| Sensor Status and Details | 137 |
| Sensor Filters | 140 |
| Bypass Reasons | 141 |

8 Uninstalling Sensors from Endpoints 144

| | |
|-----------------------------------------------------------------------------|-----|
| Uninstall Sensors from the Endpoint by using the Carbon Black Cloud Console | 144 |
| Require Codes to uninstall Sensors at an Endpoint | 145 |
| Uninstall a Linux Sensor from an Endpoint | 146 |
| Uninstall a 3.5+ macOS Sensor from an Endpoint | 146 |
| Uninstall a pre-3.5.1 macOS Sensor from an Endpoint | 146 |
| Uninstall a Windows sensor from an Endpoint | 147 |
| Uninstall Windows Sensors from an Endpoint by using Group Policy | 147 |
| Enable SCCM to Uninstall a Windows Sensor from an Endpoint | 148 |
| Delete Deregistered Sensors from Endpoints | 148 |

9 Managing Sensors for VM Workloads 149

| | |
|------------------------------------------------------------|-----|
| Installing Sensors on VM Workloads | 149 |
| Prepare Your Workloads Environment for Sensor Installation | 150 |

- Install Sensors on VM Workloads 151
- Update Sensors for Workloads from the Console 153
- Update Linux Sensors on Workloads through the Command Line 153
- Uninstall Linux Sensors from Workloads 154
- Uninstall Windows Sensors from Workloads 155
- Delete Deregistered Sensors from Workloads 155

10 Managing Kubernetes Sensors 156

- Set Up the Kubernetes Sensor 156
- Upgrade the Kubernetes Sensor 158
- Edit a Kubernetes Cluster 159
- Delete a Kubernetes Cluster 160
- Kubernetes Cluster Status 161

11 Signature Mirror Instructions 162

- Mirror Server Hardware Requirements 162
- Signature Mirror Instructions for Linux 162
- Signature Mirror Instructions for Windows 164

12 Configuring Carbon Black Cloud Communications 167

- Configure a Firewall 167
 - Disable CURL CRL CHECK 170
- Configure a Proxy 171
 - Connection Mechanism Precedence 172
 - Configure a Proxy for Windows after Sensor Installation 173
 - Configure a Proxy for Linux (all Sensor Versions) 174
 - Configure a Proxy for Linux (Sensor Versions 2.11.1+) 175
 - macOS Proxy Server Information 176

Preface

This guide provides installation and configuration instructions for Carbon Black Cloud Sensors.

You can install a Carbon Black Cloud sensor on Windows, macOS, and Linux endpoints, and on endpoints in VDI environments. The sensor provides data from the endpoints to Carbon Black Cloud analytics. You can also secure VMware workloads and Kubernetes cluster workloads by using the Carbon Black Cloud.

Intended Audience

This documentation provides sensor installation, update, and uninstall instructions for administrators, incident responders, and others who will operate the Carbon Black Cloud.

Staff who manage Carbon Black Cloud activities should be familiar with operating systems, web applications, installed software, desktop infrastructure (especially in-house procedures for software roll-outs, patch management, and anti-virus software maintenance), and the effects of unwanted software.

Getting Started with Sensor Installation

1

You can install a Carbon Black Cloud sensor on Windows, macOS, and Linux endpoints, and on endpoints in VDI environments. The sensor provides data from the endpoints to Carbon Black Cloud analytics.

The following instructions describe how to install sensors on endpoints. To install and manage sensors on workloads, see [Chapter 9 Managing Sensors for VM Workloads](#).

Method 1: Invite Users to Install Sensors on Endpoints

- Invited users receive an email that contains an installation code; each invited user installs the sensor directly on an endpoint. This method is not available for Linux sensors.
- This method is useful for installing sensors to a small number of endpoints.

Method 2: Install the Sensor on the Endpoint by using the Command Line or Software Distribution Tools

- The command line method allows for small-scale deployments and testing.
- A scripted or automated method installs the sensor by using software distribution tools. This method is useful when installing sensors across a large number of endpoints.

This chapter includes the following topics:

- [Before you Install Sensors on Endpoints](#)
- [Method 1: Invite Users to Install Sensors on Endpoints](#)
- [Method 2: Install the Sensor on the Endpoint by using the Command Line or Software Distribution Tools](#)

Before you Install Sensors on Endpoints

Make sure that endpoints meet the operating environment requirements (OER) for the Carbon Black Cloud products that you have purchased.

See the following VMware Carbon Black Cloud Sensor Operating Environment Requirements:

- [Windows Sensor \(on Windows Desktop\) OER](#)

- [Windows Sensor \(on Windows Server\) OER](#)
- [Linux Sensor OER](#)
- [macOS Sensor OER](#)

Note Some sensor names contain the product name “CB Defense.” This is correct: the same sensors apply for all Carbon Black Cloud products.

Before you install sensors on endpoints, read the following topics. Set up your AV exclusions, and configure your environment for proxy and firewall settings (see also [Chapter 12 Configuring Carbon Black Cloud Communications](#)).

About Sensor Groups and Policy Assignments

Each sensor is assigned a policy that determines what policy rules apply to the sensor.

By default, each new sensor is assigned the Standard policy unless one of the following conditions applies:

- You define an alternate policy during a command line installation.
- You have previously created sensor groups, the installed sensor matches a sensor group's criteria, and the target policy is not the Standard policy.

All the sensors in the sensor group receive an automatic assignment to a policy, which is based on the metadata that is associated with the sensor and the criteria that you define. This capability requires the following (or later) sensor versions:

- Windows sensors v3.1
- macOS sensors v3.2
- Linux sensors v2.5

You cannot define the policy during a direct user installation; however, you can change the policy to which a sensor is assigned after its installation.

Note Policy assignments do not apply to the Audit and Remediation Standalone product.

Local Scan Settings

The local scan feature is only available for Windows sensors 2.0 and later. It is not available for the Audit and Remediation Standalone product, Linux sensors, or macOS sensors.

For more information about Local Scan Settings for Windows, see [Local Scan Settings and the AV Signature Pack](#). To configure local scan settings in the console, see "Configure Local Scan Settings" in the *VMware Carbon Black Cloud User Guide*.

Setting Antivirus Exclusion Rules

You can create antivirus (AV) exclusion rules, including those specific to various endpoint platforms.

To run as usual, other AV products require custom rules.

If you use other security products, create the following exclusions for the Carbon Black Cloud sensor:

Linux

| |
|-----------------------|
| /var/opt/carbonblack/ |
| /opt/carbonblack/ |

macOS

| |
|------------------------------------------------------------|
| /Applications/Confer.app/ |
| /Applications/VMware Carbon Black Cloud |
| /Library/Application Support/com.vmware.carbonblack.cloud/ |
| /Library/Extensions/CbDefenseSensor.kext |

Windows Folders

| |
|-----------------------------|
| C:\Program Files\Confer\ |
| C:\ProgramData\CarbonBlack\ |

Windows Files

| | | |
|-----------------------------------------------------|----------------------------------------------------------|--------------------------------------------------------------|
| C:\Windows\System32\drivers\cti file.sys | C:\Windows\System32\drivers\ct inet.sys | C:\Windows\System32\drivers\cbe lam.sys |
| C:\Windows\system32\drivers\cbd isk.sys | C:\windows\system32\CbAMSI.dll | C:\windows\system32\ctiuser.dll |
| C:\windows\syswow64\CbAMSI.dll | C:\windows\syswow64\ctiuser.dl l | C:\Windows\Syswow64\ctintev.dll |
| C:\Program Files\Confer\BladeRunner.exe | C:\Program Files\Confer\CbNativeMessaging Host.exe | C:\Program Files\Confer\RepCLI.exe |
| C:\Program Files\Confer\RepMgr.exe | C:\Program Files\Confer\RepUtils.exe | C:\Program Files\Confer\RepUx.exe |
| C:\Program Files\Confer\RepWAV.exe | C:\Program Files\Confer\RepWmiUtils.exe | C:\Program Files\Confer\RepWSC.exe |
| C:\Program Files\Confer\Uninstall.exe | C:\Program Files\Confer\VHostComms.exe | C:\Program Files\Confer\Blades\LiveQuery\o squeryi.exe |
| C:\Program Files\Confer\scanner\scanhost.e xe | C:\Program Files\Confer\scanner\upd.exe | |

Set Antivirus Exclusion Rules

Use this procedure to create AV exclusion rules, including those specific to various endpoint platforms.

Note Some security vendors may require a trailing asterisk (*) to signify all directory contents.

Procedure

- 1 On the left navigation pane, click **Enforce > Policies**.
- 2 Select the policy.
- 3 Click the **Prevention** tab and expand **Permissions**.
- 4 Click **Add application path**.
- 5 Enter the AV's recommended file/folder exclusions from the security vendor.
- 6 Set the operation attempt **Performs any API operation** to **Bypass**.
- 7 To apply the changes, click **Confirm** and then click **Save**.

Method 1: Invite Users to Install Sensors on Endpoints

This method is useful when you have a small number of sensors to install, or when software distribution tools are not available. This method is not available for Linux sensors.

The installation code will expire after seven (7) days.

Important The user on the endpoint must have administrator privileges to install the sensor.

Note With the release of the Windows 3.6 sensor, you can supply either the installation code or the company code to install the sensor.

Invite Users to install Sensors

You can invite users to install sensors on their endpoints.

Procedure

- 1 Sign in to the Carbon Black Cloud console.
- 2 On the navigation bar, click **Inventory** and then click **Endpoints**.
- 3 Click **Sensor Options** and then click **Send installation request**.
- 4 Add a single user or multiple users. To add multiple users, type a comma-separated list of email addresses and then click **Send**.

Results

Users receive an email invitation that contains the installer download link and a unique single use installation code. The installation code expires after one week. If the installation code expires, follow the procedure [Send a new Installation Code](#)

The users should follow the instructions in the email to install the sensor. In the email, end users will click on the appropriate OS installer link to download the sensor.

Note We recommend that you inform users in advance that you're sending the email invitation. In the advance notification, tell the users which version to download (32-bit or 64-bit). The 32-bit variant of the sensor does not run on a 64-bit version of Windows. Instruct the users that they should copy/paste the installation code into a plain text editor, and then copy/paste that entry into the installer. Copy/pasting the installation code directly from the console does not always work properly.

If the user is installing a sensor version prior to 3.0, they must use the legacy 6-digit code instead of the extended 3.0+ installation code. You can find the 6-digit code on the Endpoints page; expand the user for whom the request is being made, and provide them with the listed v1-v2 code.

Send a new Installation Code

If installation codes have expired, you can follow these steps to send new installation codes to users.

Procedure

- 1 Sign in to the Carbon Black Cloud console.
- 2 On the navigation bar, click **Inventory** and then click **Endpoints**.
- 3 Search for and select the sensors that have expired installation codes.
- 4 Click **Take Action** and click **Send new installation code**.

Method 2: Install the Sensor on the Endpoint by using the Command Line or Software Distribution Tools

You can install sensors on the command line, or by using a scripted or automated method such as Group Policy or systems management tools.

The latter method is useful when you are installing sensors across a large number of endpoints.

Note Sensors automatically try to detect proxy settings during initial installation. This should be tested. If the automatic detection does not succeed, you must define the parameters to include the proxy IP address and port in the MSI command line. See [Configure a Proxy](#).

Important You must have administrator privileges to install the sensor.

Follow these procedures in the order listed:

- 1 [Obtain a Company Registration Code](#)
- 2 [Download Sensor Kits](#)
- 3 [Installing Sensors on Endpoints](#)

Obtain a Company Registration Code

A company registration code is required to register new sensors.

Procedure

- 1 Sign in to the Carbon Black Cloud console.
- 2 On the navigation bar, click **Inventory** and then click **Endpoints**.
- 3 Click **Sensor Options** and click **Company codes**.
- 4 Read the notification about "...generating a new code invalidates the previous code and cannot be undone" and check the box acknowledging that fact.

View Company Codes



Registration Code

Use your company code to install sensors by software distribution system or imaging

macOS sensor v3.x+ | Windows sensor v3.x+ | Linux sensor v2.x+

HWP8KSY1H8POENPSN9V3F48NPSN3C7



I understand that generating a new code invalidates the previous code and cannot be undone

Generate New Code

▶ [macOS sensor v1.x - 2.x](#) | [Windows sensor v1.x - 2.x](#)

Deregistration Code

Your organization does not have a deregistration code. Creating one allows you to uninstall any sensor in your organization



I understand that generating a new code invalidates the previous code and cannot be undone

Generate New Code

Close

- 5 If the company code has not already been assigned, under **Registration Codes**, click the **Generate New Code** button.

Results

You can also generate a company deregistration code to be required for uninstalling sensors directly at the endpoints.

Take note of the generated codes so that you can supply them during the installation. We recommend that you copy/paste the codes into a plain text editor and then copy/paste them from that source.

Note For 3.0 and later Windows or macOS sensor versions, the length of the company registration code is extended.

Use the company registration code that is specified as 3.0 to install all 3.0 and later Windows and macOS sensors, and use the 1.x — 2.x code to update Windows or macOS sensors prior to version 3.0. The process of supplying the code during sensor install remains the same. You must update any software distribution tools or any existing installation scripts to use the extended codes.

Use the code that is specified for 3.0 and later sensors to install Linux sensors.

You can change the company registration code. If you install sensors using a specific company registration code and then change the code and install sensors using the new code, the old sensors will continue to operate. Installed sensors are unaffected. Only new installation packages must use the new code.

Download Sensor Kits

You must download a sensor kit that matches the operating system of the endpoint.

Procedure

- 1 Sign in to the Carbon Black Cloud console.
- 2 On the navigation bar, click **Inventory** and then click **Endpoints**.
- 3 Click **Sensor Options** and click **Download sensor kits**.

4 Select the appropriate sensor kit version and click the link to download it.

Download Sensor Kits
×

Learn more about sensors from: [Supported Operating Systems](#), [Sensor Release Notes](#), and the [Sensor Installation Guide](#)

| OS | SENSOR VERSION | ACTION |
|----------------------------------|------------------------------|------------------------------|
| Windows 64-bit | 3.8.0.535 ▼ | Download Kit |
| Windows 32-bit | 3.8.0.535 ▼ | Download Kit |
| macOS ⓘ 10.14 - 10.15, 11, 12 | 3.4.4.51 ▼ | Download Kit |
| RHEL / CentOS / Oracle Linux | 2.13.1.933911 ▼ | Download Kit |
| Ubuntu & Debian | 2.13.1.933911 ▼ | Download Kit |
| SUSE / SLES 12 & 15 | 2.13.1.933911 ▼ | Download Kit |
| Amazon Linux 2 | 2.13.1.933911 ▼ | Download Kit |

Close

Installing Sensors on Endpoints

Sensor installation on endpoints varies by operating system and environment.

See the following sections for specific sensor installation instructions:

- [Chapter 2 Installing Linux Sensors on Endpoints](#)
- [Chapter 3 Installing macOS Sensors on Endpoints](#)
- [Chapter 4 Installing Sensors on Endpoints in a VDI Environment](#)
- [Chapter 5 Installing Windows Sensors on Endpoints](#)

Installing Linux Sensors on Endpoints

2

This section describes how to install Linux sensors from the command line.

Important Before you begin the processes described here, read [Chapter 1 Getting Started with Sensor Installation](#). It contains highly relevant information to help you succeed in your sensor installation.

Before you can install sensors, you must perform the following steps:

[Obtain a Company Registration Code](#)

[Download Sensor Kits](#)

The sensor kit is a .tgz with the format `cb-psc-sensor-<DISTRO>-<BUILD-NUMBER>.tgz`.

With the release of the Carbon Black Cloud v2.5.0 Linux sensor, Audit and Remediation and Enterprise EDR are supported on the Linux platform. The Carbon Black Cloud Linux sensor is highly modularized. It can support independent runtime enablement of Enterprise EDR and Audit and Remediation. You can manually customize the installer package to install only desired features. To install Audit and Remediation only, see [Customizing the Carbon Black Cloud Linux feature selection](#).

To configure a proxy for a Linux installation, see [Configure a Proxy for Linux \(all Sensor Versions\)](#).

Note If the company registration code contains special characters (!, #, *, \$, etc.) and is not quoted, the installation will immediately terminate. Double quotation marks are not an acceptable substitute to single quotes.

This chapter includes the following topics:

- [Unpack the Agent](#)
- [Verify the Unpacked Tar-ball Contents](#)
- [Prerequisites for Linux 4.4+ Kernels for Linux Sensor Versions 2.10+](#)
- [About the Linux Sensor cfg.ini File](#)
- [Linux Installer Command Line Parameters](#)
- [Install a Linux Sensor on a Single Endpoint](#)
- [Install a Linux Sensor on an Endpoint using the RPM/DPKG Installer](#)

- [Install a Linux Sensor on an Endpoint that Automatically Registers the First Time it is Started](#)

Unpack the Agent

The first step in installing a Linux sensor on an endpoint is to unpack the agent.

Procedure

- 1 Create a root-owned temporary install directory on the endpoint; do not use a shared folder such as `/tmp` or `/var/tmp`:

```
$ sudo mkdir cb-psc-install
```

- 2 Extract the contents of the installer package into the temporary directory you created. Replace `cb-psc-sensor-<DISTRO>-<BUILD-NUMBER>.tgz` with the filename of the installer package.

```
$ sudo tar -C cb-psc-install -xzf cb-psc-sensor-<DISTRO>-<BUILD-NUMBER>.tgz
```

Note In regards to the `-xzf` option, the `z` is optional.

Verify the Unpacked Tar-ball Contents

With the release of the 2.11.2 Linux sensor, digital-integrity verification of all tar-ball contents is enabled.

Perform the following steps to verify integrity after you unpack the TGZ and before you install the sensor.

Prerequisites

You need two tools that are usually pre-installed on Linux:

- GnuPG package (for `/usr/bin/gpg` tool)
- SHA256 checksum tool: `/usr/bin/sha256sum`

In addition, you must download the VMware Carbon Black public key, `public.asc`.

Procedure

- 1 To create a `public.asc.gpg` file, download the VMware Carbon Black public key as [public.asc](#) and then dearmor it.

```
gpg --dearmor public.asc
```

- 2 To verify the included `manifest.sha256` file with the public key, perform the following step. This step creates a `trustdb.gpg` file, which can be safely ignored.

Note In the following example output, the "Good signature" line validates the manifest. The WARNING lines can be ignored. The Signature date is the TGZ signing date.

```
$ gpg --no-default-keyring --homedir . \
--keyring public.asc.gpg \
--verify manifest.sha256.asc manifest.sha256
```

Example output:

```
gpg: WARNING: unsafe permissions on homedir '/tmp/cb-psc-install'
gpg: Signature made Wed Jun  9 01:49:05 2021 IST
gpg:          using RSA key 485BB0DF6AC57704
gpg: /tmp/cb-psc-install/trustdb.gpg: trustdb created
gpg: Good signature from "bit9build (bit9cs) <support@bit9.com>" [unknown]
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: 1853 62D1 D591 FDFA 0C64 7B58 485B B0DF 6AC5 7704
```

- 3 Check the integrity of the unpacked files: `$ sha256sum -c manifest.sha256`

```
blades/bladesUnpack.sh: OK
blades/cb-psc-lq-0.9.8200-8200-blade.tar.gz: OK
blades/cb-psc-th-0.9.8200-8200-blade.tar.gz: OK
cb-psc-sensor-2.11.2-545096.el6.x86_64.rpm: OK
cb-psc-sensor-2.11.2-545096.el7.x86_64.rpm: OK
cb-psc-sensor-2.11.2-545096.el8.x86_64.rpm: OK
install.sh: OK
```

- 4 Check for unexpected files extracted from the TGZ. You should see the files listed in the verified `manifest.sha256`, `public.asc`, `public.asc.gpg`, `trustdb.gpg`, and the two manifest files. The existence of additional files in the directory indicate that the TGZ was tampered.

Prerequisites for Linux 4.4+ Kernels for Linux Sensor Versions 2.10+

Prior to installing the sensor, the underlying BPF implementation requires the Linux kernel headers for the active kernel to be installed.

To verify that headers are installed, run the following command:

```
cat /boot/config-$(uname -r) | grep CONFIG_IKHEADERS
```

A result of either `CONFIG_IKHEADERS=m` or `CONFIG_IKHEADERS=y` means that you do not need to install any headers for BPF.

You can check the running kernel version by running the following command: `$ uname -r`

Note Secure Boot is not supported by the CBC Linux installer at this time because the kernel module is not signed. Before installation, disable Secure Boot or sign the kernel module using your preferred method. Attempting installation without doing the former will result in the sensor entering bypass mode upon installation.

For CentOS, RHEL, Oracle RHCK or Amazon Linux

- To check whether the kernel headers are installed (any user can run this):

```
$ yum list kernel-devel-$(uname -r)
```

- To install the necessary kernel headers:

```
$ sudo yum install -y kernel-devel-$(uname -r)
```

- When properly installed, the required kernel headers are located under

```
$ /usr/src/kernels/$(uname -r)/include/
```

If the kernel headers package cannot be found

Linux distributions regularly update the kernel package and might not keep the old kernel headers package in their package repos. If this happens, the easiest solution is to update the system to the latest kernel and then rerun the kernel headers install command.

To update the kernel to the latest version and install kernel headers, run the following commands (this requires a reboot):

```
$ sudo yum update kernel kernel-devel
```

```
$ reboot
```

For Oracle UEK

- To check whether the kernel headers are installed (any user can run this):

```
$ yum list kernel-uek-devel-$(uname -r)
```

- To install the necessary kernel headers:

```
$ sudo yum install -y kernel-uek-devel-$(uname -r)
```

- When properly installed, the required kernel headers are located under

```
$ /usr/src/kernels/$(uname -r)/include/
```

For SUSE or OpenSUSE

- To check whether the kernel headers are installed (any user can run this):

```
$ zypper se -s kernel-default-devel | grep $(uname -r | sed "s/-default//") $
zypper se -s kernel-devel | grep $(uname -r | sed "s/-default//")
```

- The output should be like the following, where the `i+` signifies that the package is installed. If the left-hand column is `v` or is blank, the package must be installed.

```
$ i+ kernel-default-devel | package | 4.12.14-lp150.12.25.1 | x86_64 | openSUSE-
Leap-15.0-Update
```

- To install the necessary kernel headers:

```
$ zypper install --oldpackage kernel-default-devel=$(uname -r | sed "s/-default//")
$ zypper install --oldpackage kernel-devel=$(uname -r | sed "s/-default//")
```

- When properly installed, the required kernel headers are located under

```
$/usr/src/linux-$(uname -r) | sed "s/-default//")/include/ | grep -f
```

For Debian

- To check whether the kernel headers are installed (any user can run this):

```
apt list linux-headers-$(uname -r)
```

- To install the necessary kernel headers:

```
sudo apt install linux-headers-$(uname -r)
```

- When properly installed, the required kernel headers are located under

```
/usr/src/linux-headers-$(uname -r)/include/
```

For Ubuntu

- To check whether the kernel headers are installed (any user can run this):

```
apt list linux-headers-$(uname -r)
```

- To install the necessary kernel headers:

```
sudo apt install linux-headers-$(uname -r)
```

- When properly installed, the required kernel headers are located under

```
/usr/src/linux-headers-$(uname -r)/include/
```

About the Linux Sensor `cfg.ini` File

The Linux sensor keeps its primary configuration details together with transient state in the `/var/opt/carbonblack/psc/cfg.ini` file.

The `cfg.ini` file is created when the sensor is installed. It changes while the sensor is running, and is used to manage many long term stateful processes such as software upgrades, communication configuration and state, and device registration information.

The sensor normally reads the `cfg.ini` file one time upon startup and writes it one or more times when the sensor needs to update its information. Therefore, the `cfg.ini` file should only be edited while the sensor is stopped. Modifications done while the sensor is running are likely to be overwritten by the sensor's next update of the file, and in any case are not visible to the sensor until its next startup. It is advisable to plan what changes to make to reduce the sensor downtime that occurs while editing the file.

The `install.sh` script is used to install the sensor on an endpoint (see [Install a Linux Sensor on a Single Endpoint](#)). When running this script, you can set `cfg.ini` fields by using the `--prop` option of that script. For example, the following would set the email address for this sensor:

```
./install --prop 'EmailAddress=bill@example.com'
```

Tip In this example, the parameter is enclosed in single quotation marks. This is not strictly required in this case, but it is a good practice.

Table 2-1. Supported Install.sh `cfg.ini` (`--prop`) Options

| Option | Value | Notes |
|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CompanyCode | String value Navigate to Endpoints > Sensor Options > Company Codes to access or create a new Company Code. The Company Code should be enclosed in single quotes. | The CompanyCode identifies the company that owns this machine. Across an organization, all machines should have the same values for each of these fields. These fields are set during sensor installation, by providing a registration code to the <code>install.sh</code> script. These should not be changed by the user, but may be copied to new instances as part of VDI management. |
| GroupName | String value Always enclose this value with quotes if the policy name (group name) includes spaces. | Optional policy name assignment. This field sets the Policy value for this endpoint. This affects what rulesets are applied to this sensor. This can be used to pre-set the policy used by the sensor at install time. Note It may be easier to manage this in the Carbon Black Cloud console; the backend might change this field depending on changes made in the console. This assignment can also be set during installation: <code>./install.sh --groupname 'SensorGroupName'</code> |
| EmailAddress | This can be set to any email address. | This is the point of contact for administering this sensor. The provided address is visible in the Endpoints page in the console. |

Table 2-1. Supported Install.sh cfg.ini (--prop) Options (continued)

| Option | Value | Notes |
|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ProxyServer | server:port | You can set the ProxyServer field to direct sensor network traffic through a proxy server (such as a `squid` proxy server.) You can specify the server IP address. |
| ProxyPemFile | Provide the full file path and file name of the PEM file. For example: <code>./install.sh --proxy 1.2.3.4:3129 --prop ProxyPemFile=/path/path/my-pem-file.pem.</code> | The PEM file is used to connect to some proxies that use certificate-based authentication. The PEM file is only used if the proxy server is also set. If there is no PEM file, the proxy server connection will be attempted without authentication. |

Linux Installer Command Line Parameters

The following command line parameters are supported by the `install.sh` install utility script starting with Linux sensor 2.12 and later. Parameter values must always be enclosed in single quotes.

To view all command line parameters, run the command together with the `-h` parameter.

Table 2-2.

| Parameter | Description |
|-------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| <no parameter> | Only installs the sensor on the endpoint. Does not register or start the sensor. |
| <code>-b --bypass</code> | Enable bypass mode (disabled protection) immediately after installation. |
| <code>'<Company_Code>'</code> | Registers and starts the sensor. |
| <code>-d --disable-lr</code> | Disable Live Response for the sensor during start. Live Response is enabled by default. |
| <code>-g --groupname '<POLICY_NAME>'</code> | Assign the policy to which the sensor will be added during start. |
| <code>-h --help</code> | Displays all command line options. |
| <code>-p --proxy '<PROXY_SERVER:PORT>'</code> | Preferred proxy server and port. |
| <code>-r --register</code> | This option skips starting the agent and only registers the sensor during installation. Company code must be passed when using this option. |

Install a Linux Sensor on a Single Endpoint

You can install a Carbon Black Cloud Linux sensor on a single endpoint by following this procedure.

Procedure

- 1 Extract the contents of the installer package into a temporary directory.
- 2 Install and register the sensor by running the following command; replace '<COMPANY_CODE>' with your company registration code:

```
sudo cb-psc-install/install.sh '<COMPANY_CODE>'
```

For Linux sensor versions 2.11.1 onwards, you can specify proxy server details while installing and registering the sensor through command line. Replace the '<COMPANY_CODE>' with your company registration code:

```
sudo cb-psc-install/install.sh [-p|--proxy 'proxyhost:proxyport'] '<COMPANY_CODE>'
```

For Linux sensor versions 2.6.0 onwards, you can pass optional parameters from the `cfg.ini` file to the install script. For a full list of parameters and additional information about `cfg.ini`, see [About the Linux Sensor cfg.ini File](#).

Note `-p|--proxy` is an optional parameter available for Linux sensor versions 2.11.1 onwards. It passes proxy server details for the endpoint to communicate with backend. You can specify the IP address or hostname as part of `proxyhost`.

Note The Linux sensor only supports a HTTP non-authenticated proxy server through `cfg.ini`.

Table 2-3.

| Proxy Type | IP Format | FDQN Format |
|------------|----------------|------------------|
| HTTP | ip:port | fdqn:port |
| HTTP | http://ip:port | http://fdqn:port |

Example `Cfg.ini` settings:

```
[customer]
ProxyServer=proxy.example.com:3128
```

Install a Linux Sensor on an Endpoint using the RPM/DPKG Installer

You can install a Linux sensor on an endpoint by using this method.

Procedure

- 1 Extract the contents of the installer package into a temporary directory.

- 2 Install the RPM/DEB package.

RPM:

```
$ sudo rpm -i cb-psc-install/cb-psc-sensor-<BUILD-NUMBER>.x86_64.rpm
```

DEB:

```
$ sudo dpkg -i cb-psc-install/cb-psc-sensor-<BUILD-NUMBER>.x86_64.deb
```

- 3 Install the blades.

```
$ sudo cb-psc-install/blades/bladesUnpack.sh
```

- 4 Update the `cfg.ini` file with the v3.x+ company registration code.

```
$ sudo /opt/carbonblack/psc/bin/cbagentd -d '<COMPANY_CODE>'
```

- 5 Start the agent.

For CentOS/RHEL 6:

```
$ service cbagentd start
```

For all other distributions:

```
$ systemctl start cbagentd
```

Install a Linux Sensor on an Endpoint that Automatically Registers the First Time it is Started

By using this method, the Linux sensor registers the first time it starts up.

Procedure

- 1 Extract the contents of the installer package into a temporary directory.

- 2 Use the `install.sh` script to install the agent, but do not provide a company code.

```
$ sudo cb-psc-install/install.sh
```

- 3 Update the `cfg.ini` file with the v3.x+ company code.

```
$ sudo /opt/carbonblack/psc/bin/cbagentd -d '<COMPANY_CODE>'
```

Results

Note The sensor is configured to register when the sensor starts up. This can occur on the next system boot or by restarting the agent. When the agent starts, the sensor will register itself with the Carbon Black Cloud backend.

Installing macOS Sensors on Endpoints

3

This section introduces ways to install macOS sensors on endpoints.

Important Before you begin the processes described here, read [Chapter 1 Getting Started with Sensor Installation](#). It contains highly relevant information to help you succeed in your sensor installation.

If you use [Method 1: Invite Users to Install Sensors on Endpoints](#) to install the macOS sensor 3.6.1 or later, an **Advanced** option allows you to enable FIPS. Instruct your users not to enable FIPS. This feature is not supported for commercial instances of the Carbon Black Cloud.

Before you can install sensors, you must perform the following steps:

[Obtain a Company Registration Code](#)

[Download Sensor Kits](#)

Note For VMware Workspace One instructions, see [Deploying VMware Carbon Black Cloud Sensor with Workspace ONE UEM](#).

This chapter includes the following topics:

- [Approve the Kernel Extension \(macOS 10.13 – macOS 11\)](#)
- [Approving the System Extension and Network Extension for macOS 11+](#)
- [Full Disk Access Requirement for the macOS Sensor](#)
- [Restart Requirements for macOS 10.15+](#)
- [Special Considerations for the macOS Sensor on Big Sur](#)
- [Manually Install and Approve the Sensor on macOS 11+](#)
- [macOS Sensor Command Line Install](#)
- [Deploying macOS Sensors on Big Sur and Later by using Jamf Pro](#)
- [Validate a Healthy System Extension Sensor through RepCLI](#)
- [Address the Extension Warning Post-install](#)
- [macOS Services, Utilities, and Uninstaller](#)
- [Installing macOS Sensors on Endpoints by using Workspace ONE UEM](#)

Approve the Kernel Extension (macOS 10.13 – macOS 11)

For macOS v3.1 sensor installations on macOS 10.13, High Sierra requires initial KEXT approval of the product kernel extension by administrative policy or user.

This requirement is enforced by Apple. It applies to all third-party products that have a driver component. The sensor requires KEXT approval regardless of the previous KEXT approval status.

Note For macOS Big Sur, user KEXT approval is only part of the requirement; MDM approval is required for KEXT deployment on macOS Big Sur.

Carbon Black recommends that you pre-configure endpoints with pre-approved drivers by using MDM policy, netboot, or pre-configured images. This approach simplifies sensor installation, especially during a command line installation. A CLI message occurs during the install, and requires the `- kext` flag to skip and finish the install.

If drivers are not pre-approved before sensor installation, the behavior is as follows:

Command line installation: Installation finalizes and returns success, but logs a warning to installation logs. Because drivers cannot load, the sensor enters Bypass state and reports this state to the cloud. After KEXT is approved, the sensor recovers within one hour and enters the full protection state.

Direct installation is handled similarly to a command line installation, with two differences: (1) sensor installation displays a dialog message that requests the user to approve the KEXT by using system preferences; (2) installer stalls for up 10 minutes to give the user the opportunity to approve the KEXT.

Manually Approve the KEXT

This procedure lets you manually approve the KEXT.

Procedure

- 1 Install the sensor on the endpoint.
- 2 In **System Preferences**, in the **Security & Privacy** pane, click the **General** tab.
- 3 Authenticate as Administrator.
- 4 Click the **Allow** button for **System software from developer “Carbon Black” was prevented from loading**. The installer will finish running and load the sensor.

Approve the KEXT via MDM

Use this procedure to approve KEXT using an MDM.

Procedure

- ◆ Specify the Apple Team ID and KEXT bundle in your configuration profile.
 - Apple Team ID: 7AGZNQ2S2T

- KEXT Bundle ID: `com.carbonblack.defense.kext`

See [How to approve Mac Sensor 3.0 KEXT for Install/Upgrade](#) and Apple Technical Note TN245.

Approving the KEXT via MDM for Big Sur

The easiest way to distribute the necessary MDM payload to approve the KEXT is to upload the `MDM-KEXT-approval.mobileconfig` file, which is located in the mounted DMG of the installer in the docs folder.

You can also recreate the attached mobileconfig in your MDM tool by specifying the Apple Team ID and the KEXT Bundle ID in your Kernel Extension configuration profile:

- Apple Team ID: 7AGZNQ2S2T
- KEXT Bundle ID: `com.carbonblack.defense.kext`

To allow the KEXT to load on MacOS Big Sur, the OS either requires a local action from an admin to approve the KEXT after install or a customized reboot command from your MDM to rebuild the Kernel Cache.

Your MDM must support custom XML to use the following method. If your MDM provider does not support custom XML, use the local approval method to run the KEXT.

The easiest way to distribute the necessary MDM command is to upload the `MDM-KEXT-reboot-command.xml` file, which is found in the mounted DMG of the installer in the docs folder. This XML file should be uploaded as a Custom Command and sent to endpoints after KEXT install. The target machine will reboot without warning; this distribution method is a temporary workflow until MDM providers update their reboot protocols to support `RebuildKernelCache`. This command is here:

```
<dict>
  <key>RebuildKernelCache</key>
  <true/>
  <key>KextPaths</key>
  <string>/Library/Extensions/CbDefenseSensor.kext</string>
  <key>RequestType</key>
  <string>RestartDevice</string>
</dict>
```

Identify Devices with Sensors that do not support the Operating System or need KEXT Approval

These search procedures show you which sensors are running on unsupported operating systems or are not KEXT approved.

Procedure

- 1 Sign in to the Carbon Black Cloud Console.
- 2 On the navigation bar, click **Inventory** and then click **Endpoints**.

- 3 Change the **Status** filter to **All**, and type the following search query:

```
sensorStates:UNSUPPORTED_OS
```

- 4 Use the following search query to help identify devices with sensors that do support the operating system, but with sensor KEXT or System Extension not approved:

```
sensorStates:DRIVER_LOAD_NOT_GRANTED
```

Approving the System Extension and Network Extension for macOS 11+

Beginning with macOS 11 (Big Sur), the sensor utilizes a System Extension and Network Extension (web content filter) for user space operation. In order to suppress client-side notifications to approve their operation, the System Extension and Network Extension should be pre-approved via MDM whenever possible.

For manual installation and sensor approval, see [Manually Install and Approve the Sensor on macOS 11+](#).

Approve the System Extension via MDM

Use this procedure to manually create the correct mobileconfig in your MDM.

Procedure

- ◆ Specify the Apple Team ID and System Extension bundle Identifier in your Allowed System Extension configuration profile:
 - System Extension Types: `Allowed System Extensions`
 - Apple Team ID: `7AGZNQ2S2T`
 - System Extension Bundle ID: `com.vmware.carbonblack.cloud.se-agent.extension`

The Workspace One configuration should look like the following:

System Extensions
Controls restrictions and settings for System Extensions loading on macOS 10.15 and later.

User Override
If enabled, users can approve additional system extensions that are not explicitly allowed by this policy.

Allow User Overrides ☒

Allowed System Extension Types
Allow all or some system extension types to load. Team Identifier rule takes precedence over global settings.

| Team Identifier* | Drivers | Endpoint Security | Network |
|------------------|--------------------------|--------------------------|--------------------------|
| * | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

[+ ADD SYSTEM EXTENSION TYPE](#)

Allowed System Extensions
Allow a specific set of extensions to always load. Either ID is optional, but both can be provided.

| Team Identifier | Bundle Identifier |
|-----------------|------------------------------------------------------------------|
| 7AGZNQ2S2T | com.vmware.carbonblack.cloud <input checked="" type="checkbox"/> |

[+ ADD SYSTEM EXTENSION](#)

The JAMF configuration should look like the following:

☒ Allow users to approve system extensions

Allowed Team IDs and System Extensions

Display Name
VMware Carbon Black

System Extension Types
Allowed System Extensions

Team Identifier
7AGZNQ2S2T

ALLOWED SYSTEM EXTENSIONS
com.vmware.carbonblack.cloud.se-agent.extension

Approve the Network Extension Component of the System Extension via MDM

Use this procedure to grant the System Extension the ability to Filter Network Content via a Web Content Filter configuration profile.

After creating this profile, the profile should be signed to enable distribution via MDM.

Procedure

- ◆ The fields should be completed exactly as follows. Copy and paste for accuracy.

In the General payload:

- **Payload Scope:** `System`

In the Web Content Filter payload:

- **Filter Type:** `Plug-In`
- **Plug-In Bundle ID:** `com.vmware.carbonblack.cloud.se-agent`
- Check **Enable Socket Filtering**
 - **Filter Data Provider System Extension Bundle ID (macOS):**
`com.vmware.carbonblack.cloud.se-agent.extension`
 - **Filter Data Provider Designated Requirement (macOS):** `identifier "com.vmware.carbonblack.cloud.se-agent.extension" and anchor apple generic and certificate 1[field.1.2.840.113635.100.6.2.6] /* exists */ and certificate leaf[field.1.2.840.113635.100.6.1.13] /* exists */ and certificate leaf[subject.OU] = "7AGZNQ2S2T"`
- Check **Enable Packet Filtering (macOS)**
 - **Filter Packet Provider System Extension Bundle ID (macOS):**
`com.vmware.carbonblack.cloud.se-agent.extension`
 - **Filter Packet Provider Designated Requirement (macOS):** `identifier "com.vmware.carbonblack.cloud.se-agent.extension" and anchor apple generic and certificate 1[field.1.2.840.113635.100.6.2.6] /* exists */ and certificate leaf[field.1.2.840.113635.100.6.1.13] /* exists */ and certificate leaf[subject.OU] = "7AGZNQ2S2T"`

Full Disk Access Requirement for the macOS Sensor

As part of user data security enhancements in macOS 10.14.5 and above, you must approve access to protected user and application data.

This requirement is in addition to kernel extension approval, and does not replace that process.

Applications can be granted access to app data (such as photos, contacts, and calendars), protected services and devices (such as the microphone or camera), or user data (such as mail, cookies, and Safari history) via the **Security & Privacy System Preferences** pane. Access is granted by enabling individual access, or by allowing full disk access. For the macOS sensor to operate at full functionality on an endpoint that is running macOS 10.14.5+, the sensor must have full disk access.

To be completely effective, the macOS sensor must be granted access to protected user and application data. This can be done manually on each endpoint, or for quicker and more consistent endpoint management, these settings can be managed through the creation and deployment of a mobile device management (MDM) profile.

Manually Grant the pre-3.5.1 Sensor Full Disk Access

Use this procedure to grant full disk access to pre-3.5.1 sensor versions.

Procedure

- 1 Go to the **Security & Privacy System Preferences** section and click the **Privacy** tab.
- 2 After you are authenticated as an Administrator, scroll down to the **Full Disk Access** section and click the **Plus (+)** button to add an application. Select **/Applications/Confer.app**.
- 3 Restart Confer.app. After the restart, `com.carbonblack.defense.ui` will appear in the allowed applications list.

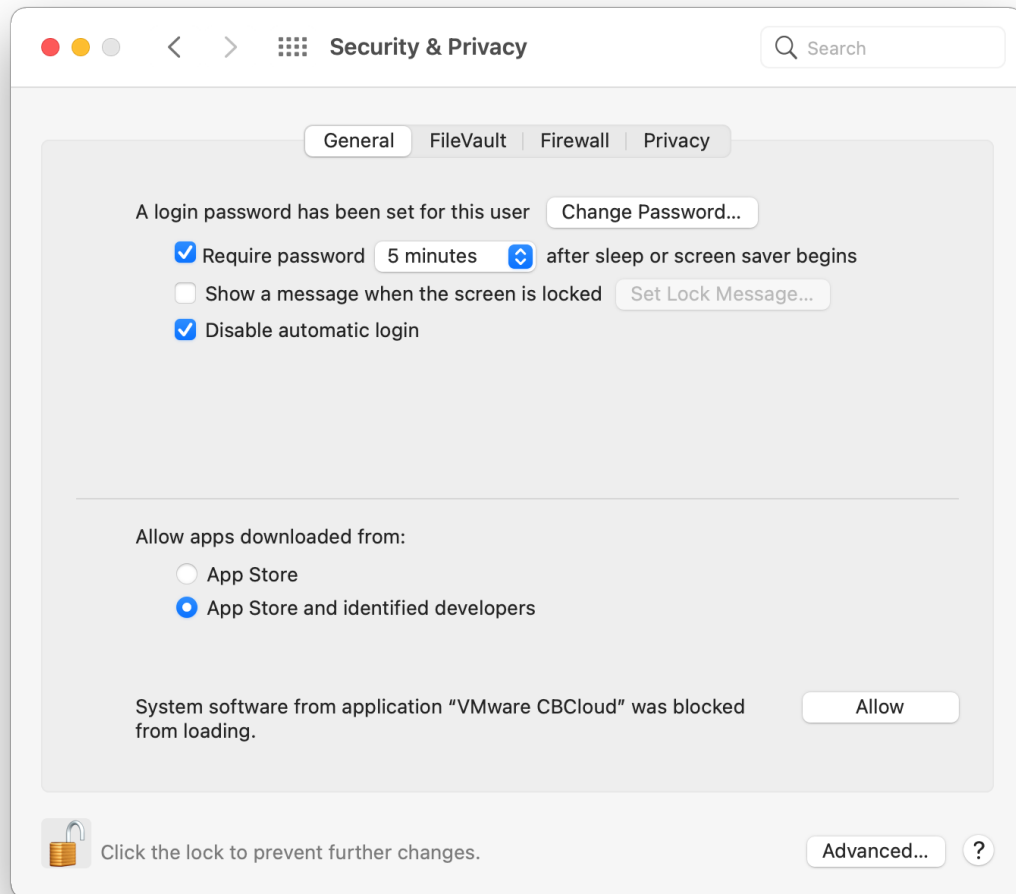
Manually Grant the 3.5.1 or Later Sensor Full Disk Access

Use this procedure to manually grant Full Disk Access to macOS sensors.

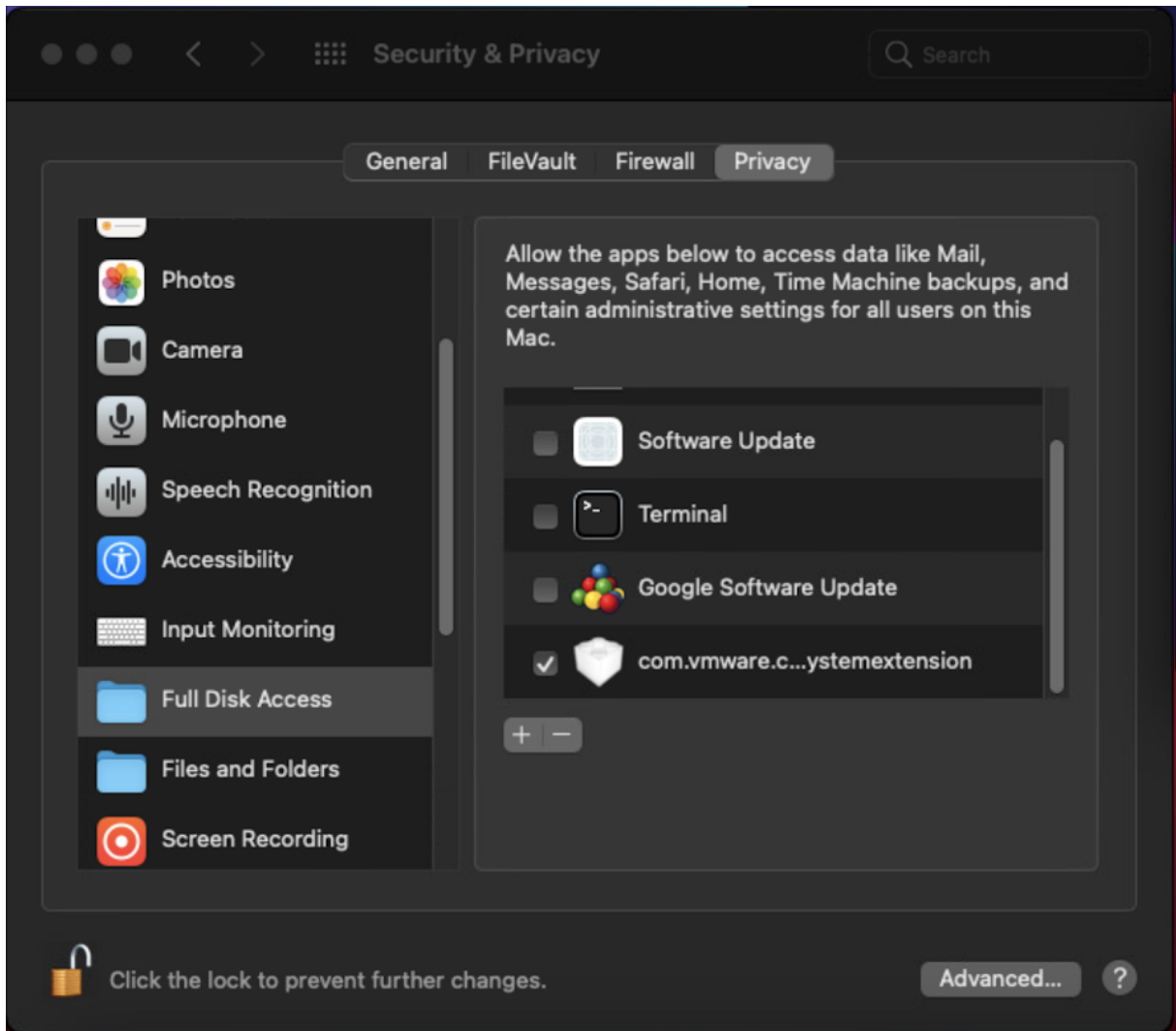
Procedure

- 1 In System Preferences, open the **Security & Privacy** pane and scroll down to **Full Disk Access**. Click the **Lock** icon to edit the settings.

- 2 In the **Full Disk Access** window, locate `com.vmware.carbonblack.cloud.se-agent.extension.systemextension` and select its checkbox.



- 3 Open a **Finder** window and go to `/Applications/VMware Carbon Black Cloud/`.
- 4 Drag *only* the following folder items into the **Full Disk Access** list:
 - LiveQuery bundle
 - repmgr bundle
 - uninstall bundle
 - VMware CBCloud



Important An Apple-acknowledged bug in the **Full Disk Access** pane prevents bundles from displaying in the window. It will appear as if the drag was not successful. This is not the case.

- 5 Verify that Full Disk Access is successfully enabled. It takes approximately one hour for RepCLI to successfully report Full Disk Access changes. To verify the change immediately, restart the endpoint.
- 6 Open a Terminal window and run one of the following commands (requires admin password):

```
cd /Applications/VMware\ Carbon\ Black\ Cloud/repcli.bundle/Contents/MacOS/
sudo ./repcli status
```

OR

```
sudo /Applications/VMware\ Carbon\ Black\ Cloud/repcli.bundle/Contents/MacOS/repcli status
```

RepCLI displays a sensor information report, including Full Disk Access status:

```

bit9qa@macos-20D91 MacOS % sudo ./repcli status
General Info:
  Sensor Version: 3.5.2.64
  Kernel Type: System Extension
  System Extension: Running
  Kernel File Filter: Connected
  Background Scan: Standard Scan
  Sensor Restarts: 8
  Last Reset: not set
Full Disk Access Configurations:
  Repmgr: Configured Manually
  System Extension: Configured Manually
  OSQuery: Configured Manually
  Uninstall Helper: Configured Manually
  Uninstall UI: Not Configured

```

Grant the 3.51+ Sensor Full Disk Access via MDM

The easiest way to distribute the necessary Privacy Preference payload is to upload the `MDM-privacyconfig.mobileconfig` file, which is in the mounted DMG of the installer in the docs folder.

The following steps recreate the mobileconfig in your MDM.

These instructions were created using Apple documentation and were validated in Jamf PRO and WorkspaceONE UEM using sensor version 3.5.0.30. Field names, values, and functionality vary depending on the MDM framework or sensor version.

Granting an application full disk access is accomplished via a Privacy Preferences payload. The Carbon Black Cloud Sensor requires five identifiers in this Privacy payload.

Procedure

- ◆ The fields should be completed exactly as follows. Copy and paste for accuracy.

Identifier: `com.vmware.carbonblack.cloud.daemon`

Identifier Type: `Bundle ID`

Code Requirement:

```

identifier "com.vmware.carbonblack.cloud.daemon" and anchor apple generic
    and certificate 1[field.1.2.840.113635.100.6.2.6] /* exists */ and
    certificate leaf[field.1.2.840.113635.100.6.1.13] /* exists */ and
    certificate leaf[subject.OU] = "7AGZNQ2S2T"

```

App or Service: `SystemPolicyAllFiles`

Access: `Allow`

Identifier: `com.vmware.carbonblack.cloud.osqueryi`

Identifier Type: Bundle ID

Code Requirement:

```
identifier "com.vmware.carbonblack.cloud.osqueryi" and anchor apple generic
    and certificate 1[field.1.2.840.113635.100.6.2.6] /* exists */ and
    certificate leaf[field.1.2.840.113635.100.6.1.13] /* exists */ and
    certificate leaf[subject.OU] = "7AGZNQ2S2T"
```

App or Service: SystemPolicyAllFiles

Access: Allow

Identifier: com.vmware.carbonblack.cloud.se-agent.extension

Identifier Type: Bundle ID

Code Requirement:

```
identifier "com.vmware.carbonblack.cloud.se-agent.extension" and anchor apple generic
    and certificate 1[field.1.2.840.113635.100.6.2.6] /* exists */ and
    certificate leaf[field.1.2.840.113635.100.6.1.13] /* exists */ and
    certificate leaf[subject.OU] = "7AGZNQ2S2T"
```

App or Service: SystemPolicyAllFiles

Access: Allow

Identifier: com.vmware.carbonblack.cloud.uninstall

Identifier Type: Bundle ID

Code Requirement:

```
identifier "com.vmware.carbonblack.cloud.uninstall" and anchor apple generic
    and certificate 1[field.1.2.840.113635.100.6.2.6] /* exists */ and
    certificate leaf[field.1.2.840.113635.100.6.1.13] /* exists */ and
    certificate leaf[subject.OU] = "7AGZNQ2S2T"
```

App or Service: SystemPolicyAllFiles

Access: Allow

Identifier: com.vmware.carbonblack.cloud.uninstallerui

Identifier Type: Bundle ID

Code Requirement:

```
identifier "com.vmware.carbonblack.cloud.uninstallerui" and anchor apple
    generic and certificate 1[field.1.2.840.113635.100.6.2.6] /*
exists */ and
    certificate leaf[field.1.2.840.113635.100.6.1.13] /* exists */ and
    certificate leaf[subject.OU] = "7AGZNQ2S2T"
```

App or Service: SystemPolicyAllFiles

Access: Allow

Restart Requirements for macOS 10.15+

Beginning with macOS 10.15, a system reboot is required in the following scenarios.

- Installing the sensor for the first time on macOS 10.15+.
- Installing or upgrading the sensor in KEXT mode on macOS 11.

Endpoints that require a reboot report that state on the Dashboard or Endpoints page; search for `sensorStates:DRIVER_INIT_REBOOT_REQUIRED` on the Endpoints page to find 10.15+ devices that are in bypass mode and require a reboot.

Special Considerations for the macOS Sensor on Big Sur

This section describes considerations for installing macOS sensors on Big Sur.

Install a KEXT-enabled Sensor on Big Sur

Use this procedure to install a KEXT-enabled Sensor on Big Sur. On macOS 11, the attended installer defaults to installing a System Extension sensor. To install into KEXT mode, we recommend using the `cbcloud_install_unattended.sh` install script, which is found in the mounted DMG of the sensor installer in the docs folder.

Note [Approving the KEXT via MDM for Big Sur](#) is required for KEXT operation on macOS Big Sur. If you do not have an MDM tool, do not attempt KEXT install on macOS Big Sur.

A new `-k` flag in `cbcloud_install_unattended.sh` signifies a KEXT sensor install. This flag also works during an update. See [macOS Command Line Parameters](#).

For Kernel Extensions (legacy System Extensions) to run on macOS Big Sur, Apple has added two new restrictions for new installs or updates:

- Kernel Extensions must be pre-approved via MDM.
- Kernel Extensions must be approved manually, and the OS requires a reboot after install.

Procedure

- 1 Run the `cbcloud_install_unattended.sh` script. Your mount point may be slightly different than what is shown here:

```
sudo /Volumes/CBCloud-3.5.1.19/docs/cbcloud_install_unattended.sh -i /Volumes/  
CBCloud-3.5.1.19/CBCloud\ Install.pkg -c [Company Registration Code] -k
```

- 2 Before the install finishes, a window appears stating that a System Extension has been updated. Approve this prompt in the **Security & Privacy** pane of **System Preferences**. The KEXT must be pre-approved using an MDM provider prior to installation or this alert will be suppressed by the operating system. See [Approving the KEXT via MDM for Big Sur](#) for more information on this process. For manual Full Disk Access instructions, see [Manually Grant the 3.5.1 or Later Sensor Full Disk Access](#).
- 3 Restart the OS to finish installing the new KEXT.

Switch the macOS Sensor Kernel Type on Big Sur during Update

On Big Sur, to switch between kernel types during an update, run the `cbcloud_install_unattended.sh` script with either the `-k` or `-e` flag.

The `-k` flag will force a Kernel Extension sensor. The `-e` flag will force a System Extension sensor. See [macOS Command Line Parameters](#).

Your mount point might be slightly different than what is shown

```
here: sudo /Volumes/CBCloud-3.5.1.19/docs/cbcloud_install_unattended.sh -i /Volumes/
CBCloud-3.5.1.19/CBCloud\ Install.pkg -c [Company Registration Code] -k
```

Toggle between Kernel Extension and System Extension in Big Sur

We highly recommend that you perform the toggle command after you have configured MDM for both Kernel Extension and System Extension in Big Sur.

Your organization's deregistration code is required to run the toggle command. In the Carbon Black Cloud console, go to **Inventory > Endpoints > Sensor Options > View Company Codes**. In this context, the code does not uninstall anything and is used as an administrative code to enable the RepCLI tool. For more information about RepCLI, see [Managing Sensors by using RepCLI](#) in the user guide.

Procedure

- ◆ Perform one of the following procedures:
 - To toggle from System Extension to Kernel Extension:
 - a Run the following command:


```
sudo /Applications/VMware\ Carbon\ Black\ Cloud/repcli.bundle/Contents/
macOS/repcli setsensorkext [deregistration code]
```
 - b The KEXT must be manually approved. A window will appear stating that a System Extension has been updated. Approve this prompt in the **Security & Privacy** pane of **System Preferences**.
 - c Restart the OS.

- To toggle from Kernel Extension to System Extension:

- a Run the following command:

```
sudo /Applications/VMware\ Carbon\ Black\ Cloud/repcli.bundle/Contents/  
macOS/repcli setsensorsysex [deregistration code]
```

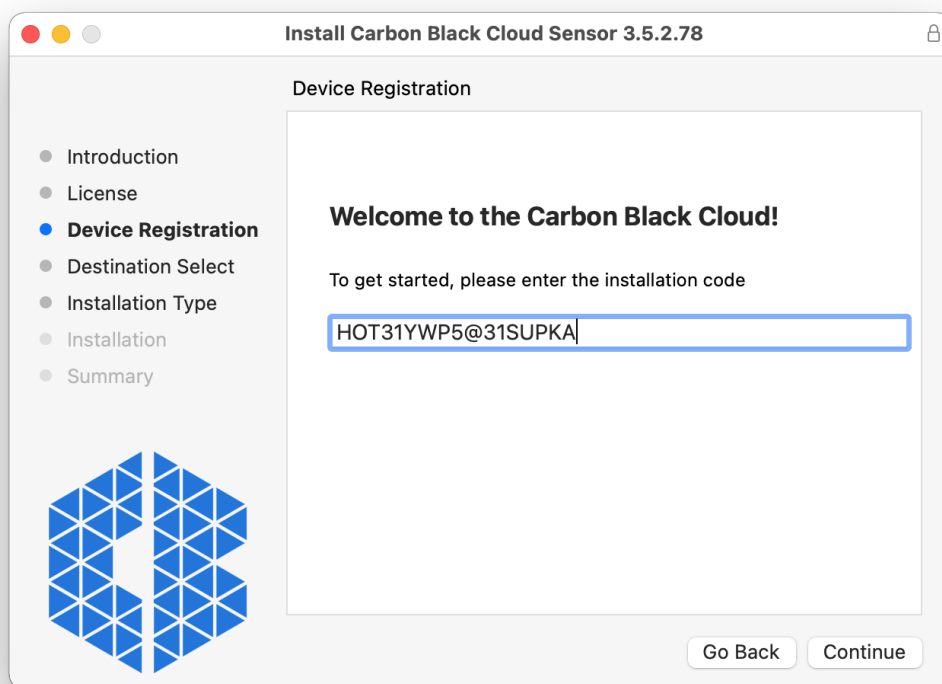
- b Restart the OS.

Manually Install and Approve the Sensor on macOS 11+

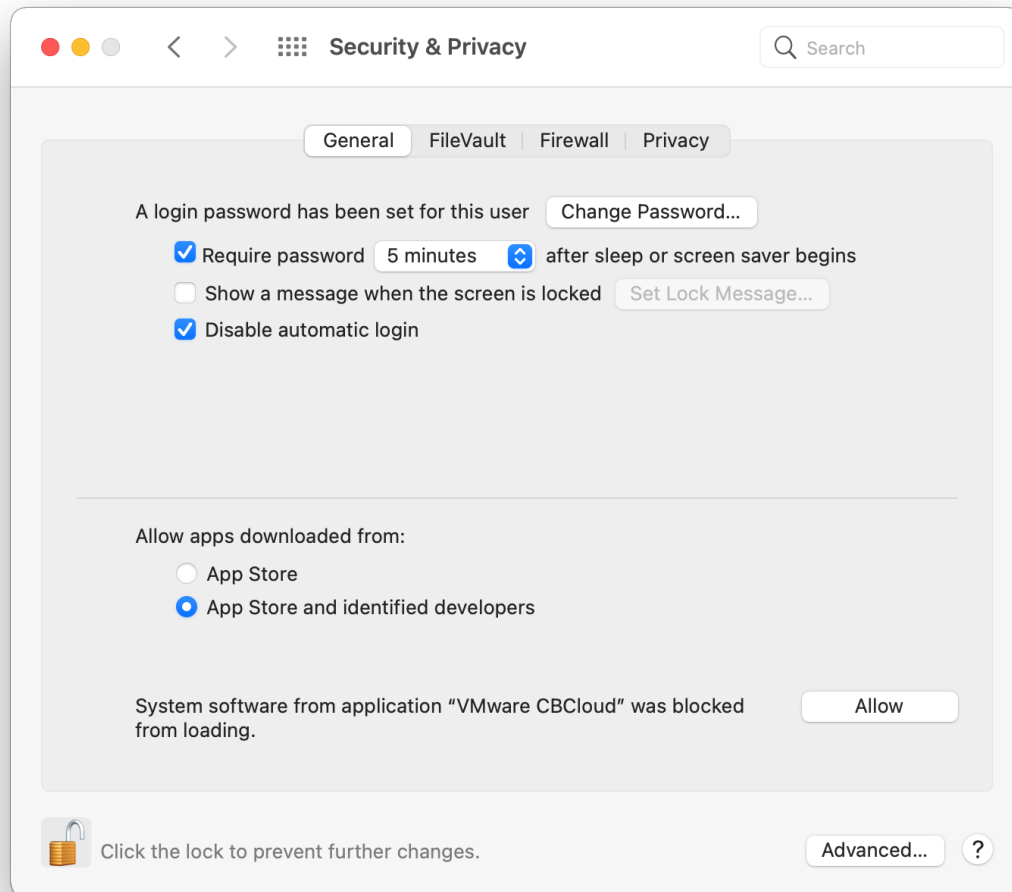
This article describes how to manually install and approve the Carbon Black Cloud sensor on macOS Big Sur (macOS 11+).

Procedure

- 1 Start the Carbon Black Cloud installer. The installer will request access to your Desktop folder. Click **OK**.
- 2 Enter the sensor installation code. If the installation code is entered incorrectly, an error message will state that the installer cannot communicate with the Carbon Black Cloud. Check the installation code and try again.



- 3 When the installer finishes running, a message will notify you that you need to approve the VMware Carbon Black Cloud system extension. Click **Open Security Preferences** to open the **Security & Privacy** pane.
- 4 On the **General** tab in the **Security & Privacy** pane, a notification indicates that the VMware CBCloud system extension was blocked from loading. Enter your administrator password to unlock the pane and then click the **Allow** button next to the notification.

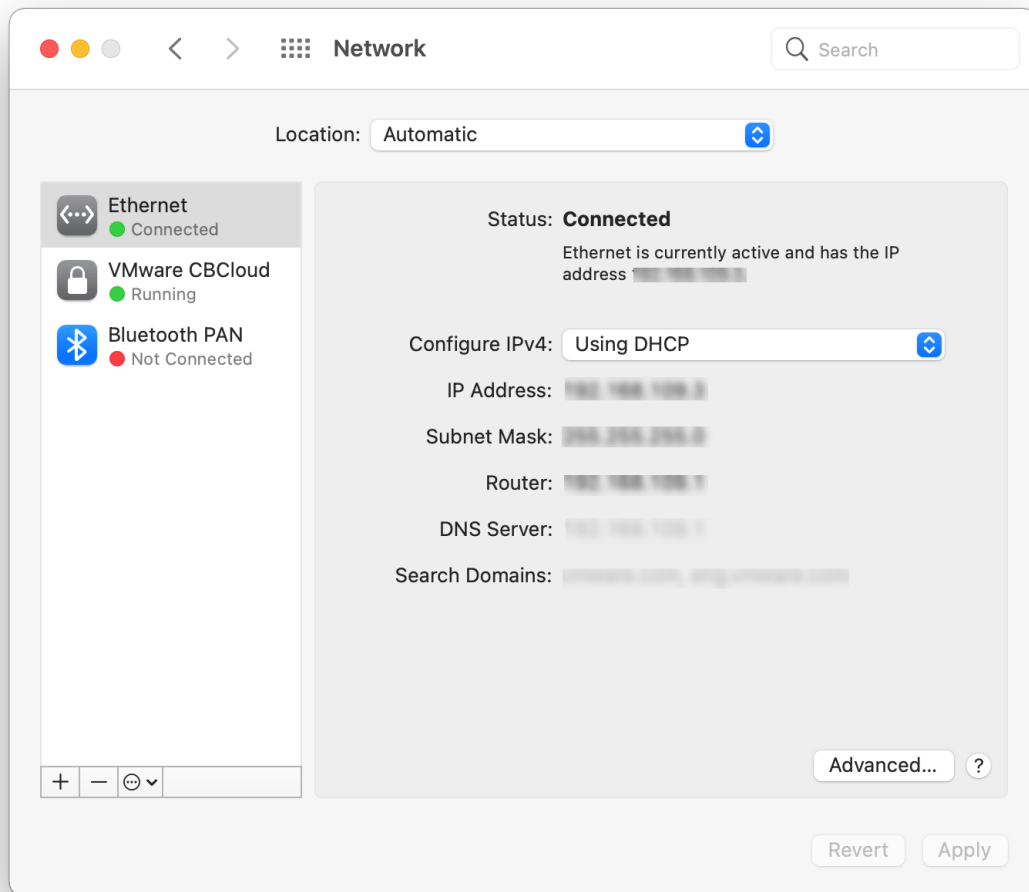


Note The notification will persist until the system extension is approved, even if the endpoint is restarted. Until it is approved, the sensor will be in bypass mode.

- 5 After the system extension is approved, another notification will state that the VMware Carbon Black Cloud sensor wants to filter network content. This is required for the sensor to report network events to the Carbon Black Cloud console. Click **Allow**.

Note If you click **Don't Allow** instead of **Allow**, network events will not be sent to the backend. You can restart this prompt by running the `CBCloud.app` in the `/Applications/VMware Carbon Black Cloud` folder.

- 6 To verify that the network extension is approved, go to the **Network** pane. An entry for *VMware CBCloud* will appear as a network interface.



macOS Sensor Command Line Install

The `CBCloud Install.pkg` and `ccloud_install_unattended.sh` scripts are part of the macOS sensor release and are embedded in the CB Defense DMG. Both files are required for command line installations on macOS endpoints.

Note Instructions on how to create custom packages for software distribution tools are beyond the scope of this article. Carbon Black provides generic instructions on how to install the `CBCloud Install.pkg` payload on the command line, with the help of the `ccloud_install_unattended.sh` utility script. You can adapt these instructions to a software distribution tool.

macOS utility script

The utility script can be used in the following ways:

- As-is (passed command line options to customize the install process).
- Modified to hard-code the install options and simplify the installation.
- Used as an example or guide on how to create a custom script.

A common installation method is to use the utility script as-is, push the script and the PKG payload onto the target device (both files can be bundled in a custom package), and then execute the utility script.

Extract and Prepare the macOS Install Files

Before you can install sensors, you must extract and prepare the macOS install files.

Procedure

- 1 Click **CB Cloud DMG** or mount it by using system tools. DMG is mounted to the `/Volumes/CBCloud-X.X.X.X` directory (where X.X.X.X refers to the sensor version).
- 2 Alternatively, use the `hdiutil` command to mount the downloaded sensor release disk image; for example: `hdiutil attach /path/to/CBCloud_Installer_mac_X.X.X.X`
- 3 Extract the `CBCloud Install.pkg` file from the mounted volume `/Volumes/CBCloud-X.X.X.X` directory. The `.pkg` file is the sensor installer payload.
- 4 Extract the `ccloud_install_unattended.sh` utility script from the `/Volumes/CBCloud-X.X.X.X/docs/` directory.
- 5 The mounted volume can be unmounted because it is not needed for the remainder of the steps. You can unmount it by using `Finder`, or by running the following command: `hdiutil eject /Volumes/CBCloud-X.X.X.X`
- 6 Use the extracted `CBCloud Install.pkg` and `ccloud_install_unattended.sh` files to create a custom package that is compatible with your software distribution tool or install the two files directly onto the target macOS device.

Results

Note `cbcloud_install_unattended.sh` and the `CBCloud Install.pkg` payload must be extracted from the same major and minor version of released DMG file to ensure compatibility between the utility and the installer payload. If the two files do not originate from the same release, the installation might fail.

Typically, the extracted `cbcloud_install_unattended.sh` and the `CBCloud Install.pkg` files are pushed to the target server endpoints. They can be used to create a custom installation bundle that is compatible with a specific software distribution tool.

Note You must always wrap the company registration code in single quotation marks. Double quotation marks are not an acceptable substitute to single quotes.

Perform a macOS Sensor Command Line Installation

Follow this procedure to install a macOS sensor from the command line.

Procedure

- 1 Extract the files from a sensor release DMG file.
- 2 Optionally, create a custom wrapper package bundle that is compatible with the selected software distribution tool. The custom package embeds the `CBCloud Install.pkg` file, together with a utility to set up options and start the sensor PKG installation.
- 3 Install the sensor installer on the endpoint by using the supported options.

macOS Command Line Parameters

The following common command line parameters are supported by the `cbcloud_install_unattended.sh` utility script. The parameters are passed on to the installer.

Note Parameter values must always be enclosed in single quotes.

To view all command line parameters, run the command together with the `-h` parameter.

| Parameter | Required or Optional | Description |
|----------------------------------------------|----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>-c</code> <code>COMPANY_CODE</code> | Required | Company registration code. |
| <code>-d</code> | Optional | Enter bypass mode (disabled protection) immediately after installation. You can enable protection at a later time. This mode is only recommended for test situations. |
| <code>-e</code> | Optional | Forces System Extension install on macOS Big Sur (the sensor will default to this mode on Big Sur and does not need to be explicitly specified). |
| <code>-g</code> <code>POLICY_NAME</code> | Optional | Specify a policy to which the sensor will be added. |

| Parameter | Required or Optional | Description |
|-----------------------------------|----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -h | | Displays all command line options, including advanced options that are not documented here. Refer to the built-in help in the <code>cbcloud_install_unattended.sh</code> utility script for currently supported installation options. |
| -i PKG_FILE | Required | Absolute path to the PKG installer payload. |
| -k | Optional | Forces Kernel Extension install on macOS Big Sur (pre-approvals must be in place). |
| -o | Optional | Username/email address override. Used during registration and for identifying the device. |
| -p PROXY_SERVER:PORT | Optional | Preferred Proxy server and port; for example: -p '10.5.6.7:54443' Multiple proxy servers can be provided and separated by semi-colons; for example: - p '10.5.6.8:54443;10.5.6.7:54443' If a proxy server/port are not specified but are required, the sensor will attempt proxy auto-detection. See Configure a Proxy . |
| -s | Optional | Background scan enabled ("on") or disabled ("off"). Default is enabled. Cloud policy overrides this setting. |
| -t | Optional | File upload limit (in MB). Default is no limit. |
| -u | Optional | Disable auto-update. Auto-update is enabled by default. |
| -v | Optional | Show version of this script. Major and minor versions should match the version of the Carbon Black Cloud package being deployed. |
| -x PROXY_USER:PASSWORD | Optional | Proxy credentials to use for the proxy server, if required. These apply whether the proxy server is auto-detected or specified. Example: -x 'proxy_user:proxy_password' If proxy credentials are not specified, but are required by the proxy server, the macOS sensor will attempt to detect and use proxy credentials that are stored in the keychain that match the detected or specified proxy server. |
| --skip-kext-approval-check=1 | Optional | Allows for >=3.1 sensor install/upgrade to run on macOS >=10.13 even if KEXT approval has not been done prior to the install/upgrade. KEXT approval can be deferred until after the sensor install/upgrade. Warning macOS 11.0 requires a KEXT MDM pre-approval to install a KEXT sensor. |
| --disable-live-response=1 | Optional | Disable Live Response. |
| --disable-sysextnetwork-extension | Optional | Disable network extension. Only available in System Extension mode on macOS 11 and later. Network Extension is enabled by default. |
| --disable-upgrade-jitter=1 | Optional | Disable auto-update jitter. |

Obfuscation of command line inputs

Endpoint users might input sensitive data into the command line. The obfuscation of command line inputs protects against unauthorized users accessing the data in plain text in the sensor `.log` files and the sensor databases. You can obfuscate command line inputs by using the following argument in the unattended install script: `--enable-hide-command-lines=1`

The setting enables the obfuscation of command line input in sensor `.log` files and databases. The data in the Carbon Black Cloud console is not obfuscated.

macOS Command Line Install Examples

Review the following examples for macOS command line installations.

The following commands should be on a single line.

The following examples assume that the required files are installed to the target device `/tmp/` directory.

To run a command line install with required parameters

```
sudo /tmp/cbcloud_install_unattended.sh -i '/tmp/CBCloud Install.pkg' -c 'XYZ'
```

To specify a policy for the sensor

```
sudo /tmp/cbcloud_install_unattended.sh -i '/tmp/CBCloud Install.pkg' -c 'XYZ' -g 'Monitored'
```

Deploying macOS Sensors on Big Sur and Later by using Jamf Pro

You can use Jamf Pro to deploy the Carbon Black Cloud macOS sensor on macOS systems that are running Big Sur.

We recommend that you use the latest macOS sensor version for your deployment. The configuration documented here was tested using the following versions:

- macOS Big Sur 11.4
- Jamf Pro 10.30
- macOS sensor 3.5.3.82

Note We offer this procedure as guidance only. VMware does not provide official support of Jamf software.

The basic deployment workflow is as follows:

- 1 Create and deploy a configuration profile.

- 2 Deploy a distribution policy that subsequently deploys a package that you created in Jamf Pro to a temporary location. The policy then runs a script that contains the installer package location and the company registration. Optionally, the script can install the sensor in Kernel extension mode. System Extension is the default mode.

To deploy the macOS sensor by using Jamf Pro, perform the following steps:

- 1 Download a macOS sensor kit and obtain a company registration code.
- 2 Create a package using Jamf Composer.
- 3 Obtain, prepare, and upload the installation script.
- 4 Create and deploy a configuration profile.
- 5 Create and assign smart computer groups.

Obtain and Prepare the Sensor

To obtain and prepare the sensor kit for deployment, perform the following procedure.

Procedure

- 1 Follow these steps to download the latest macOS sensor kit: [Download Sensor Kits](#).
- 2 Mount the DMG file that you downloaded in Step 1 and locate the `CBCloud Install.pkg` file.
- 3 Follow these steps to find and make note of the company registration code: [Obtain a Company Registration Code](#).

Important Do not generate a new company registration code; simply make a note of the current code.

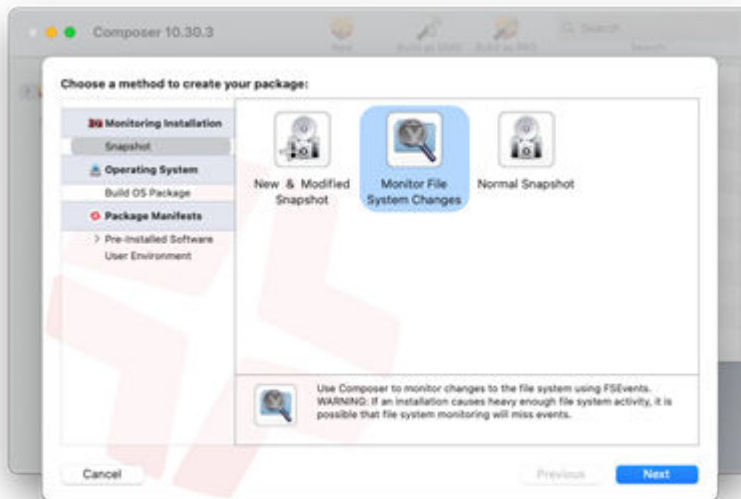
Create a Package by using Jamf Composer

Create a package for the installer in a temporary location on the endpoint.

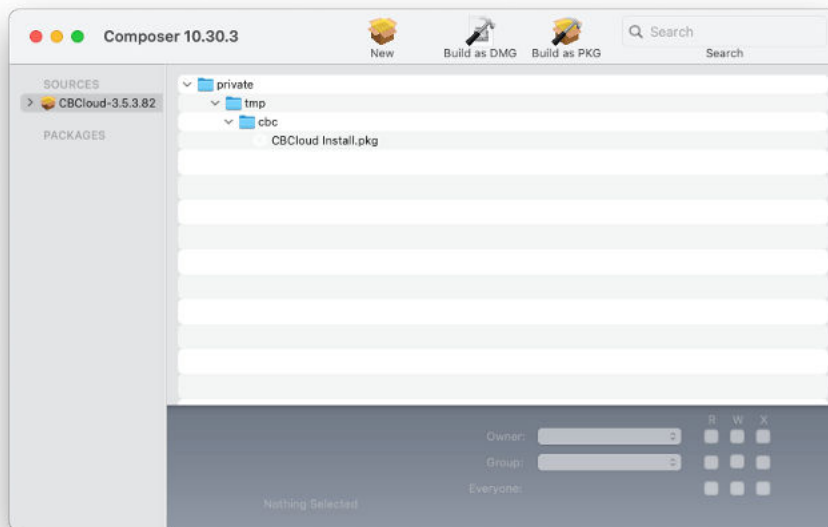
Procedure

- 1 Open Jamf Composer and go to **Preferences**. Remove `/private/tmp` from the **Exclusion List** and save the settings.

- 2 Create a new package using your preferred method - in this case, **Monitor File System Changes**. Click **Next**.



- 3 Enter an appropriate package name and click **Next**.
- 4 Create the file path `/private/tmp/cbc`. Copy the `CBCloud Install.pkg` file from the downloaded sensor kit into this directory. Click **Create Package Source** in Jamf Composer.



- 5 Make sure that `/private/tmp/cbc` is the only file path included in the package.
- 6 Click **Build as DMG** and save.

Modify the Installation Script

Modify the `cbcloud_install_unattended.sh` script for the macOS sensor deployment.

Procedure

- 1 In the mounted DMG, locate the `cbcloud_install_unattended.sh` file in the `docs` folder.
- 2 Copy `cbcloud_install_unattended.sh` to a location where it can be modified.
- 3 Open `cbcloud_install_unattended.sh` in a plain text editor and locate the following lines (beginning on line 49):

```
#options
CBC_INSTALLER=""
COMPANY_OR_USER_CODE=""
```

- 4 Set `CBC_INSTALLER` as the temporary location of the `CBCloud Install.pkg` that you established in [Create a Package by using Jamf Composer](#).
- 5 Modify the `COMPANY_OR_USER_CODE` to be the company registration code. For example:

```
CBC_INSTALLER="/private/tmp/cbc/CBCloud Install.pkg"
COMPANY_OR_USER_CODE="3TABC99SW2021"
```

Optional: If you are deploying the sensor in Kernel Extension mode, modify the following line:

```
KERNEL_TYPE=0
```

to:

```
KERNEL_TYPE=1
```

- 6 Save the changes.

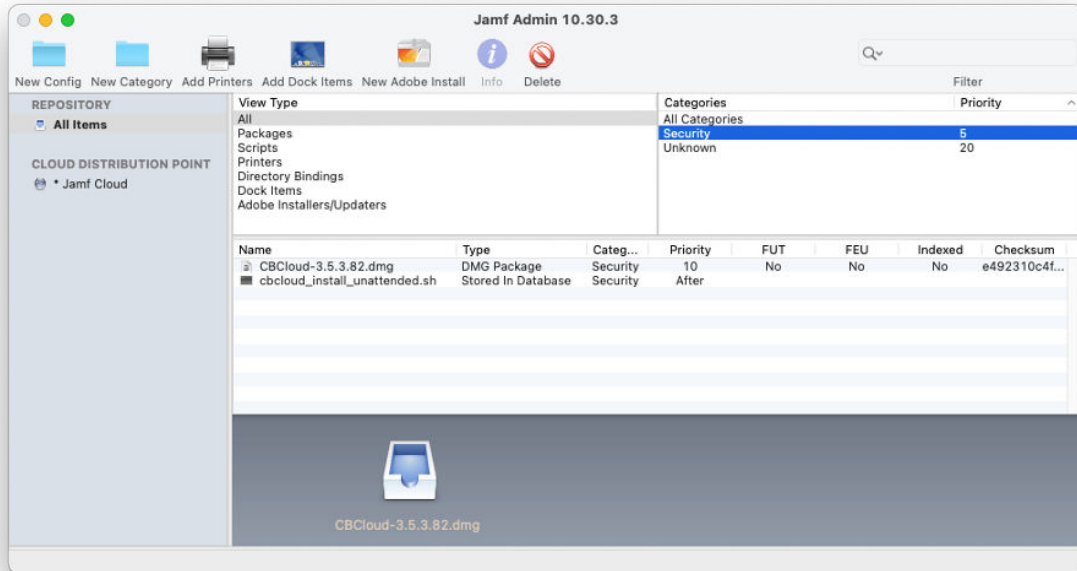
Upload macOS Sensor DMG and Installation Script to Jamf Pro

Use Jamf Admin to upload the macOS Sensor DMG and Installation Script to Jamf Pro.

Procedure

- 1 Start Jamf Admin.
- 2 Add the DMG and installation script files to the repository.

3 Set the `cbcloud_install_unattended.sh` file **Priority** to **After**. Example:



Creating a Configuration Profile

To allow the deployment of Carbon Black Cloud macOS sensors on macOS Big Sur systems, create a configuration profile. Include the System Extension or Kernel Extension payloads as required for your implementation.

Configure Configuration Profile General Settings

To configure the **General** settings in the configuration profile, define the following parameters.

Procedure

- 1 Provide a descriptive **Name** for the profile. For example, "Carbon Black Cloud Settings - System Extension".
- 2 Add a brief **Description** to the profile.
- 3 Set the **Category** for your organization.
- 4 Set the **Level** to **Computer Level**.
- 5 Set the **Distribution Method** to **Install Automatically**.

Set Privacy Preferences Policy Control in the Configuration Profile

To ensure full functionality of the macOS sensor, enter each App Access sub-payload from the following table. For all sub-payloads, the **Identifier Type** is `Bundle ID`, and the **Application or Service** is `SystemPolicyAllFiles` with **Access** set to `Allow`.

| Identifier | Code Requirement |
|--------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>com.vmware.carbonblack.cloud.daemon</code> | <pre> identifier "com.vmware.carbonblack.cloud.daemon" and anchor apple generic and certificate 1[field.1.2.840.113635.100.6.2.6] /* exists */ and certificate leaf[field.1.2.840.113635.100.6.1.13] /* exists */ and certificate leaf[subject.OU] = "7AGZNQ2S2T" </pre> |
| <code>com.vmware.carbonblack.cloud.se-agent.extension</code> | <pre> identifier "com.vmware.carbonblack.cloud.se- agent.extension" and anchor apple generic and certificate 1[field.1.2.840.113635.100.6.2.6] /* exists */ and certificate leaf[field.1.2.840.113635.100.6.1.13] /* exists */ and certificate leaf[subject.OU] = "7AGZNQ2S2T" </pre> |
| <code>com.vmware.carbonblack.cloud.osqueryi</code> | <pre> identifier "com.vmware.carbonblack.osqueryi" and anchor apple generic and certificate 1[field.1.2.840.113635.100.6.2.6] /* exists */ and certificate leaf[field.1.2.840.113635.100.6.1.13] /* exists */ and certificate leaf[subject.OU] = "7AGZNQ2S2T" </pre> |
| <code>com.vmware.carbonblack.cloud.uninstall</code> | <pre> identifier "com.vmware.carbonblack.cloud.uninstall" and anchor apple generic and certificate 1[field.1.2.840.113635.100.6.2.6] /* exists */ and certificate leaf[field.1.2.840.113635.100.6.1.13] /* exists */ and certificate leaf[subject.OU] = "7AGZNQ2S2T" </pre> |
| <code>com.vmware.carbonblack.cloud.uninstallerui</code> | <pre> identifier "com.vmware.carbonblack.cloud.uninstallerui" and anchor apple generic and certificate 1[field.1.2.840.113635.100.6.2.6] /* exists */ and certificate leaf[field.1.2.840.113635.100.6.1.13] /* exists */ and certificate leaf[subject.OU] = "7AGZNQ2S2T" </pre> |

Enable System Extension Payloads in the Configuration Profile

If you are deploying the macOS sensor in System Extension mode, include a System Extensions payload. You can optionally include both System Extension and Kernel Extension payloads to provide flexibility.

Procedure

- 1 To optionally enable users to approve extensions that are not included in this payload, select **Allow Users to approve system extensions**.
- 2 Define the following settings in **Allowed Team IDs and System Extensions**:
 - **Display Name**: Carbon Black Cloud System Extension
 - **System Extension Types**: **Allowed System Extensions**
 - **Team Identifier**: 7AGZMQ2S2T
 - **Allowed System Extensions**:

```
com.vmware.carbonblack.cloud.se-agent.extension
```

For example:

Enable Kernel Extension Payloads in the Configuration Profile

If you are deploying the macOS sensor in Kernel Extension mode (macOS 10.14 - macOS 11), include an Approved Kernel Extension payload. You can optionally include both System Extension and Kernel Extension payloads to provide flexibility.

Procedure

- 1 Select **Allow Users to approve system extensions**.
- 2 Optionally select **Allow standard users to approve legacy Kernel Extensions (macOS 11 or later)**: This selection is dependent on your environment.
- 3 Define the following settings in **Allowed Team IDs and Kernel Extensions**:
 - **Approved Team ID Display Name**: VMware Carbon Black
 - **Approved Team ID**: 7AGZMQ2S2T
 - **Approved Kernel Extension Display Name**: Carbon Black Cloud Sensor

■ Approved Kernel Extension Kernel Extension Bundle ID:

```
com.carbonblack.defense.kext
```

For example:

Approved Kernel Extensions

☒ Allow users to approve kernel extensions

☐ Allow standard users to approve legacy kernel extensions (macOS 11 or later)

Approved Team IDs and Kernel Extensions

Approved Team ID

Display Name

VMware Carbon Black

Team ID

7AGZNQ2S2T

Approved Kernel Extensions (Optional) Approve the following bundle IDs only

| DISPLAY NAME | KERNEL EXTENSION BUNDLE ID | |
|---------------------------|------------------------------|-------------|
| Carbon Black Cloud Sensor | com.carbonblack.defense.kext | Edit Delete |

Note This payload pre-approves the Kernel Extension. The user must still enable the Kernel Extension in the **Security & Privacy** section of **System Preferences** on the endpoint after you have installed the sensor. The user must then reboot the endpoint.

Set the Content Filter in the Configuration Profile

To enable the configuration deployment without requiring user approval of the network extension, create the following payload.

Procedure

1 Set the **Filter Name**:

```
VMware Carbon Black Cloud Network Extension Filter.
```

2 Set the **Identifier**:

```
com.vmware.carbonblack.cloud.se-agent
```

3 Set the **Socket Filter Bundle Identifier**:

```
com.vmware.carbonblack.cloud.se-agent.extension
```

4 Set the **Socket Filter Designated Requirement**:

```
identifier "com.vmware.carbonblack.cloud.se-agent.extension" and anchor apple generic
and certificate 1[field.1.2.840.113635.100.6.2.6] /* exists */ and certificate
leaf[field.1.2.840.113635.100.6.1.13] /* exists */ and certificate leaf[subject.OU] =
"7AGZNQ2S2T"
```

5 Set the **Network Filter Bundle Identifier**:

```
com.vmware.carbonblack.cloud.se-agent.extension
```

6 Set the **Network Filter Designated Requirement**:

```
identifier "com.vmware.carbonblack.cloud.se-agent.extension" and anchor apple generic
and certificate 1[field.1.2.840.113635.100.6.2.6] /* exists */ and certificate
leaf[field.1.2.840.113635.100.6.1.13] /* exists */ and certificate leaf[subject.OU] =
"7AGZNQ2S2T"
```

7 Save the configuration profile.

Example:

Content Filter
Settings configured: 4

Filter Name
Display name of the filter in the app and on the device

VMware Carbon Black Cloud Network Extension Filter

Required

Identifier
Identifier for the filter plug-in

com.vmware.carbonblack.cloud.se-agent

Required

Socket Filter

Socket Filter Bundle Identifier Bundle identifier of the socket filter provider system extension

com.vmware.carbonblack.cloud.se-agent.extension

Required

Socket Filter Designated Requirement Designated requirement of the socket filter provider system extension

Identifier "com.vmware.carbonblack.cloud.se-agent.extension" and anchor apple generic and certificate [field.1.2.840.113635.100.6.2.6] /* exists */ and certificate leaf[field.1.2.840.113635.100.6.1.13] /* exists */ and certificate leaf[subject.OU] = "7AGZNQ2S2T"

Required

Network Filter

Network Filter Bundle Identifier Bundle identifier of the network filter provider system extension

com.vmware.carbonblack.cloud.se-agent.extension

Required

Network Filter Designated Requirement Designated requirement of the network filter provider system extension

Identifier "com.vmware.carbonblack.cloud.se-agent.extension" and anchor apple generic and certificate [field.1.2.840.113635.100.6.2.6] /* exists */ and certificate leaf[field.1.2.840.113635.100.6.1.13] /* exists */ and certificate leaf[subject.OU] = "7AGZNQ2S2T"

Required

Create a Software Distribution Policy

In Jamf Pro, create a software distribution policy.

Procedure

- 1 Provide a descriptive policy **Display Name**.
- 2 Set the **Trigger** to **Recurring Check-in**.

Example:

The screenshot shows the 'General' tab of a policy configuration window in Jamf Pro. The 'Display Name' is 'Carbon Black Cloud Sensor 3.5.3.82 (System Extension)'. The 'Enabled' checkbox is checked. The 'Category' is set to 'Security'. The 'Trigger' section has several options: 'Startup', 'Login', 'Logout', 'Network State Change', 'Enrollment Complete', and 'Recurring Check-in'. The 'Recurring Check-in' checkbox is checked, while the others are unchecked.

General

Display Name Display name for the policy

Carbon Black Cloud Sensor 3.5.3.82 (System Extension)

☒ **Enabled**

Category Category to add the policy to

Security

Trigger Event(s) to use to initiate the policy

☐ **Startup**
When a computer starts up. A startup script that checks for policies must be configured in Jamf Pro for this to work

☐ **Login**
When a user logs in to a computer. A login hook that checks for policies must be configured in Jamf Pro for this to work

☐ **Logout**
When a user logs out of a computer. A logout hook that checks for policies must be configured in Jamf Pro for this to work

☐ **Network State Change**
When a computer's network state changes (e.g., when the network connection changes, when the computer name changes, when the IP address changes)

☐ **Enrollment Complete**
Immediately after a computer completes the enrollment process

☒ **Recurring Check-in**
At the recurring check-in frequency configured in Jamf Pro

- 3 On the **Packages** tab, select the Carbon Black Cloud DMG file that you built using Jamf Composer.
- 4 Set the **Distribution Point** to **Each computer's default distribution point**.
- 5 Set **Action** to **Install**.
- 6 On the **Scripts** tab, select the `cbcloud_install_unattended.sh` file that you modified. Set the priority to **After**. No parameter values are required.
- 7 Save the policy.

Create and Assign Smart Computer Groups

Create and assign two smart computer groups for the macOS sensor deployment.

You must set the scope of two smart computer groups to the configuration profile and software policy. This scope ensures that the configuration profile is deployed to the endpoint before you install the Carbon Black Cloud macOS sensor. If you deploy the Carbon Black Cloud sensor without having deployed the configuration profile, the user receives approval prompts. If the prompts are not approved, the sensor is not fully functional.

Note Before you can select the Carbon Black Cloud Settings Configuration Profile in the second smart group (steps 5 through 10), you must first deploy the configuration profile to a macOS endpoint and update the endpoint inventory. After Jamf Pro recognizes the installed configuration profile, the profile becomes a selectable option when you are creating the smart computer group.

Procedure

- 1 On the Jamf Pro dashboard, on the **Computers** tab, click **Smart Computer Groups** and then click **+ New**.

- 2 On the **Computer Group** tab, enter a name such as “macOS Big Sur Computers”.
 - 3 Click the **Criteria** tab and set the following criteria:
 - **Criteria: Operating System**
 - **Operator: like**
 - **Value: 11**
 - 4 Save the smart computer group.
 - 5 On the Jamf Pro dashboard, on the **Computer** tab, click **Smart Computer Groups** and then click **+ New**.
 - 6 On the **Computer Group** tab, enter a name such as “Carbon Black Cloud Settings”.
 - 7 Click the **Criteria** tab and set the following criteria:
 - **Criteria: Profile Name**
 - **Operator: Has**
 - **Value:** Click the ... menu and select the Carbon Black Cloud Settings – System Extension Configuration Profile.
 - 8 Save the smart computer group.
 - 9 For the configuration profile, assign the “**macOS Big Sur computers**” smart computer group to the scope.
 - 10 For the policy, assign the “Carbon Black Cloud Settings” smart computer group to the scope.
- After the smart computer groups are assigned, deployment begins.

Important If deploying the Kernel Extension, you must approve it after it is installed in **System Preferences>Security & Privacy**. This is an Apple-imposed requirement. This approval is not required if deploying the System Extension. We recommend that you ask the user to approve this extension by using a notification in the deployment policy.

Validate a Healthy System Extension Sensor through RepCLI

Follow this procedure to validate a healthy System Extension Sensor on Big Sur.

Procedure

- ◆ Run the following command:

```
sudo /Applications/VMware\ Carbon\ Black\ Cloud/repcli.bundle/Contents/macOS/repcli
status
```

For more information about RepCLI, see [Managing Sensors by using RepCLI](#) in the User Guide.

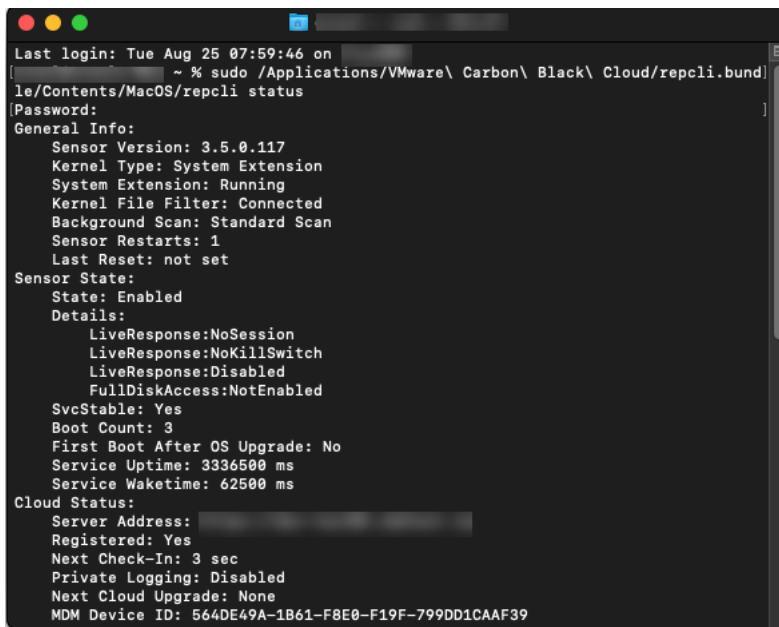
The following results are expected:

General Info

- Kernel Type: System Extension
- System Extension: Running
- Kernel File Filter: Connected

Sensor State

- State: Enabled
- SvcStable: Yes (Might take a few minutes after install to reach SvcStable)
- Cloud Status:
- Registered: Yes



```

Last login: Tue Aug 25 07:59:46 on
~ % sudo /Applications/VMware\ Carbon\ Black\ Cloud/repcli.bund
le/Contents/MacOS/repcli status
Password:
General Info:
  Sensor Version: 3.5.0.117
  Kernel Type: System Extension
  System Extension: Running
  Kernel File Filter: Connected
  Background Scan: Standard Scan
  Sensor Restarts: 1
  Last Reset: not set
Sensor State:
  State: Enabled
Details:
  LiveResponse:NoSession
  LiveResponse:NoKillSwitch
  LiveResponse:Disabled
  FullDiskAccess:NotEnabled
SvcStable: Yes
Boot Count: 3
First Boot After OS Upgrade: No
Service Uptime: 3336500 ms
Service Waketime: 62500 ms
Cloud Status:
  Server Address:
  Registered: Yes
  Next Check-In: 3 sec
  Private Logging: Disabled
  Next Cloud Upgrade: None
  MDM Device ID: 564DE49A-1B61-F8E0-F19F-799DD1CAAF39
  
```

Address the Extension Warning Post-install

A post-install warning can occur after installing a macOS sensor. This procedure resolves the problem.

Procedure

- 1 Download the installer: macOS.
- 2 When prompted to approve the CB Defense kernel extension, click **OK**.
- 3 When the **System Extension Blocked** message appears, click **Open Security Preferences**.
- 4 Click **Allow** next to **System software from developer “Carbon Black, Inc.” was blocked**.
- 5 Double-click the Carbon Black icon and copy/paste the installation code from a text editor.

macOS Services, Utilities, and Uninstaller

This section lists the macOS sensor services, utilities, and uninstaller files that reside on the endpoints.

macOS installed services for 3.5.0 and lower

- Sensor Driver Bundle: `/System/Library/Extensions/CBDefenseSensor.kext`
- Sensor Service: `/Applications/Confer.app/Contents/MacOS/repmgr`
- Sensor UI: `/Applications/Confer.app/Contents/MacOS/CBDefense`

macOS installed services for 3.5.1 and higher

- Sensor Driver Bundle: `/Applications/VMware Carbon Black Cloud/`
- Sensor data directories: `/Library/Application Support/com.vmware.carbonblack.cloud/`
- Sensor Service: `/Applications/VMware Carbon Black Cloud/repmgr.bundle/Contents/MacOS/repmgr`
- Sensor UI: `/Applications/VMware Carbon Black Cloud/CBCloudUI.bundle/Contents/MacOS/CBCloudUI`

macOS installed utilities

- Uninstaller helper: `/VMware Carbon Black Cloud/uninstall.bundle/Contents/MacOS/uninstall`
- Upgrade helper: `/VMware Carbon Black Cloud/UpgradeHelper.bundle/Contents/MacOS/UpgradeHelper`
- RepCLI: `/VMware Carbon Black Cloud/repcli.bundle/Contents/MacOS/repcli`

macos uninstaller

- 3.X: `/Applications/Confer.app/uninstall`
- 1.X sensor: `/Applications/Confer.app/uninstall.sh`
- `CLI_USERS=sid #Required, needed to interact with sensor locally`

Installing macOS Sensors on Endpoints by using Workspace ONE UEM

This section describes how to deploy the Carbon Black Cloud macOS sensor through Workspace ONE UEM.

Note The steps can vary based on the specific version of macOS, Carbon Black Cloud macOS sensor, and Workspace ONE UEM that you are using. This content was created using macOS Big Sur 11.1, Workspace ONE UEM 2101, and the Carbon Black Cloud macOS sensor version 3.5.1.19.

Extension Types

Starting with macOS 11, the Carbon Black Cloud macOS sensor (v3.5.1) operates by default in user-space by using System Extensions (user-space) instead of Kernel Extensions (KEXTs) that were used in prior versions. As a result, some functionality is temporarily unavailable when using the sensor in System Extension mode on macOS 11 and later versions. Using the sensor in KEXT mode achieves the same functionality on macOS 11 as seen in older operating systems. Differences in functionality are listed at [Carbon Black Cloud macOS Sensor Operating Environment Requirements](#).

Prerequisites

- Workspace ONE UEM with permissions to manage devices and applications
- Carbon Black Cloud console access and admin account credentials
- An endpoint running macOS to test the integration

Prepare to Install macOS Sensors

Before deploying the Carbon Black Cloud sensor for macOS on your endpoint you must obtain an installation code and download the sensor installer Carbon Black Cloud. As Workspace ONE administrator, you use the registration code to connect your endpoint to the respective Carbon Black Cloud environment tenant.

There are two methods to retrieve the sensor installer and the installation code.

- You can send a sensor installation request through the Carbon Black Cloud console. Then, you receive an email invitation that contains the installer download link and a unique single use installation code. For more details, see [Method 1: Invite Users to Install Sensors on Endpoints](#).
- You can use the company registration code and download a sensor kit that matches the operating system of the endpoint. For more details, see [Obtain a Company Registration Code](#) and [Download Sensor Kits](#).

By default, the Carbon Black Cloud macOS sensor version 3.5.1.19 and later installs System Extensions on macOS Big Sur 11.0 and later.

Deploying the Carbon Black Cloud sensor for macOS Manually with System Extensions

By default, the Carbon Black Cloud macOS sensor version 3.5.1.19 and later installs System Extensions on macOS Big Sur 11.0 and later.

As a Workspace ONE administrator, you must create a macOS device profile by using the Workspace ONE UEM console and then deploy the Carbon Black Cloud sensor for macOS with System Extensions.

Prerequisites

- Check for updates on Carbon Black Cloud sensor for macOS version 3.5.1.19 and later.
 - For updates on Carbon Black Cloud sensor for macOS version 3.6.1.10 and 3.6.2.110, see *macOS Sensor Release Notes*, part of the *Carbon Black Cloud documentation*.
 - For updates on Carbon Black Cloud sensor for macOS version 3.5.1.19 to 3.6.1.10 see [Carbon Black Cloud macOS Sensor Release Notes \(Carbon Black Cloud Community\)](#).
- Make sure you are familiar with smart groups. For information on creating smart groups with the UEM console, see *Getting Started → Console Basics → Assignment Groups*, part of the *VMware Workspace ONE UEM Console Documentation*.
- If you plan to deploy the Carbon Black Cloud sensor in KEXT mode, VMware recommends submitting the applicable kernel extension IDs for approval by Workspace ONE UEM before installing or upgrading the macOS sensor version 3.5 and later. For details, see [Approve the Kernel Extension \(macOS 10.13 – macOS 11\)](#).
- To deploy the Carbon Black Cloud sensor in System Extension mode, VMware recommends submitting the applicable system extension IDs for approval by Workspace ONE UEM before installing or upgrading the macOS sensor version 3.5 and later. For details, see [Approving the System Extension and Network Extension for macOS 11+](#).

Procedure

1 [Creating a Configuration Profile](#)

To allow the deployment of Carbon Black Cloud macOS sensors on macOS Big Sur systems, create a configuration profile. Include the System Extension or Kernel Extension payloads as required for your implementation.

2 [Manually Install and Approve the Sensor on macOS 11+](#)

After publishing the device configuration profile, manually install and approve the Carbon Black Cloud sensor on macOS Big Sur (macOS 11+).

3 [Confirm the Carbon Black Cloud Sensor Installed on the macOS Device](#)

You can use the RepCLI output to verify that the Carbon Black Cloud sensor for macOS is installed successfully on the device.

Creating a Configuration Profile

To allow the deployment of Carbon Black Cloud macOS sensors on macOS Big Sur systems, create a configuration profile. Include the System Extension or Kernel Extension payloads as required for your implementation.

Procedure

1 Configure Device Profile General Settings

To configure the **General** settings in the device configuration profile and ensure that the Carbon Black sensor has the appropriate permissions granted prior installation, define the following parameters in the Workspace ONE UEM console.

2 Enable Kernel Extension Payloads in the Configuration Profile

If you are deploying the macOS sensor in Kernel Extension mode (macOS 10.14 - macOS 11), include an approved Kernel Extension payload. You can optionally include both System Extension and Kernel Extension payloads to provide flexibility.

3 Enable System Extension Payloads in the Configuration Profile

If you are deploying the macOS sensor in System Extension mode, include a System Extensions payload. You can optionally include both System Extension and Kernel Extension payloads to provide flexibility.

4 Set Privacy Preferences Policy Control in the Configuration Profile

For the macOS sensor to operate at full functionality on an endpoint, the sensor must have full disk access on the endpoint. This payload grants the macOS sensor full disk access.

5 Set the Content Filter in the Configuration Profile

To enable the configuration deployment without requiring user approval of the network extension, configure the custom settings payload.

Configure Device Profile General Settings

To configure the **General** settings in the device configuration profile and ensure that the Carbon Black sensor has the appropriate permissions granted prior installation, define the following parameters in the Workspace ONE UEM console.

Procedure

1 Provide a descriptive **Name** for the profile.

For example, **Carbon Black Settings**.

2 Add a brief **Description** to the profile.

3 Set the **Assignment Type** to **Auto**.

4 Populate the **Smart Groups** text box with the smart groups you used for deploying the Carbon Black Cloud sensor installer to macOS Big Sur 11.0 and later.

Enable Kernel Extension Payloads in the Configuration Profile

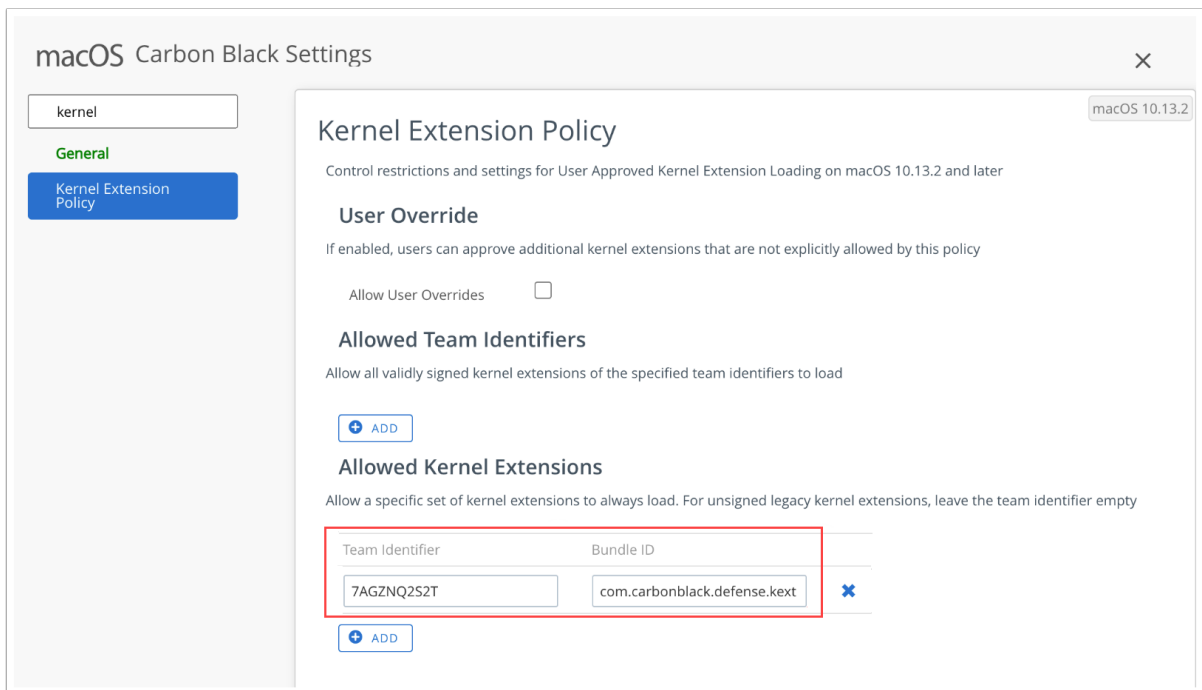
If you are deploying the macOS sensor in Kernel Extension mode (macOS 10.14 - macOS 11), include an approved Kernel Extension payload. You can optionally include both System Extension and Kernel Extension payloads to provide flexibility.

Procedure

- 1 Enter **kernel** in the search text box of the device configuration profile.
- 2 Select the **Kernel Extension Policy** payload option and click **Configure**.
- 3 Enter the Carbon Black Cloud team identifier and the Carbon Black Cloud kernel extension bundle ID.

For example:

- **Team identifier:** 7AGZNQ2S2T
- **Bundle ID:** com.carbonblack.defense.kext



Note This payload pre-approves the Kernel Extension. The user must still enable the Kernel Extension in the **Security & Privacy** section of **System Preferences** on the endpoint after you have installed the sensor. The user must then reboot the endpoint.

Enable System Extension Payloads in the Configuration Profile

If you are deploying the macOS sensor in System Extension mode, include a System Extensions payload. You can optionally include both System Extension and Kernel Extension payloads to provide flexibility.

Procedure

- 1 Enter **system** in the search text box of the device configuration profile.
- 2 Select the `System Extensions` payload option and click **Configure**.

Note If you are deploying the Sensor in Kernel Extension mode, pre-staging the System Extensions settings prepares you for a later migration from Kernel Extension to System Extensions.

- 3 Click **Add System Extension**.
- 4 Enter the Carbon Black Cloud team identifier and the Carbon Black Cloud system extension bundle ID.

For example:

- **Team Identifier:** 7AGZNQ2S2T
- **Bundle ID:** com.vmware.carbonblack.cloud.se-agent.extension

The screenshot shows the 'macOS Add a New Apple macOS Profile' window. On the left, a sidebar contains a search box with 'system' and two buttons: 'General' and 'System Extensions'. The 'System Extensions' button is selected. The main content area is titled 'System Extensions' and 'macOS 10.15'. It contains the following sections:

- User Override:** A section with a description and a checkbox for 'Allow User Overrides' which is currently unchecked.
- Allowed System Extension Types:** A section with a description and a table of extension types. The 'Team Identifier*' column has a dropdown menu showing an asterisk (*). The other columns are 'Drivers', 'Endpoint Security', and 'Network', each with an unchecked checkbox.
- Allowed System Extensions:** A section with a description and two input fields: 'Team Identifier' (containing '7AGZNQ2S2T') and 'Bundle Identifier' (containing 'com.vmware.carbonblack.cloud'). This section is highlighted with a red rectangular box.

At the bottom right of the window are two buttons: 'SAVE AND PUBLISH' and 'CANCEL'.

Set Privacy Preferences Policy Control in the Configuration Profile

For the macOS sensor to operate at full functionality on an endpoint, the sensor must have full disk access on the endpoint. This payload grants the macOS sensor full disk access.

To ensure full functionality of the macOS sensor, enter each App Access sub-payload from the following table. For all sub-payloads, the **Identifier Type** is `Bundle ID`, and the **Application or Service** is `SystemPolicyAllFiles` with **Access** set to `Allow`.

| Identifier | Code Requirement |
|--------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>com.vmware.carbonblack.cloud.daemon</code> | <pre> identifier "com.vmware.carbonblack.cloud.daemon" and anchor apple generic and certificate 1[field.1.2.840.113635.100.6.2.6] /* exists */ and certificate leaf[field.1.2.840.113635.100.6.1.13] /* exists */ and certificate leaf[subject.OU] = "7AGZNQ2S2T" </pre> |
| <code>com.vmware.carbonblack.cloud.se-agent.extension</code> | <pre> identifier "com.vmware.carbonblack.cloud.se- agent.extension" and anchor apple generic and certificate 1[field.1.2.840.113635.100.6.2.6] /* exists */ and certificate leaf[field.1.2.840.113635.100.6.1.13] /* exists */ and certificate leaf[subject.OU] = "7AGZNQ2S2T" </pre> |
| <code>com.vmware.carbonblack.cloud.osqueryi</code> | <pre> identifier "com.vmware.carbonblack.osqueryi" and anchor apple generic and certificate 1[field.1.2.840.113635.100.6.2.6] /* exists */ and certificate leaf[field.1.2.840.113635.100.6.1.13] /* exists */ and certificate leaf[subject.OU] = "7AGZNQ2S2T" </pre> |
| <code>com.vmware.carbonblack.cloud.uninstall</code> | <pre> identifier "com.vmware.carbonblack.cloud.uninstall" and anchor apple generic and certificate 1[field.1.2.840.113635.100.6.2.6] /* exists */ and certificate leaf[field.1.2.840.113635.100.6.1.13] /* exists */ and certificate leaf[subject.OU] = "7AGZNQ2S2T" </pre> |
| <code>com.vmware.carbonblack.cloud.uninstallerui</code> | <pre> identifier "com.vmware.carbonblack.cloud.uninstallerui" and anchor apple generic and certificate 1[field.1.2.840.113635.100.6.2.6] /* exists */ and certificate leaf[field.1.2.840.113635.100.6.1.13] /* exists */ and certificate leaf[subject.OU] = "7AGZNQ2S2T" </pre> |

Set the Content Filter in the Configuration Profile

To enable the configuration deployment without requiring user approval of the network extension, configure the custom settings payload.

Procedure

- 1 Enter **custom** in the search text box of the device configuration profile.
- 2 Select **Custom Settings** and click **Configure**.
- 3 Copy and paste the following custom XML for the sensor's network extension.

This setup grants System Extensions the ability to Filter Network Content by using a Web Content Filter configuration profile.

Example of a custom settings XML.

```
<dict>
  <key>FilterDataProviderBundleIdentifier</key>
  <string>com.vmware.carbonblack.cloud.se-agent.extension</string>
  <key>FilterDataProviderDesignatedRequirement</key>
  <string>identifier "com.vmware.carbonblack.cloud.se-agent.extension" and anchor apple
generic and certificate 1[field.1.2.840.113635.100.6.2.6] /* exists */ and certificate
leaf[field.1.2.840.113635.100.6.1.13] /* exists */ and certificate leaf[subject.OU] =
"7AGZNQ2S2T"</string>
  <key>FilterPacketProviderBundleIdentifier</key>
  <string>com.vmware.carbonblack.cloud.se-agent.extension</string>
  <key>FilterPacketProviderDesignatedRequirement</key>
  <string>identifier "com.vmware.carbonblack.cloud.se-agent.extension" and anchor apple
generic and certificate 1[field.1.2.840.113635.100.6.2.6] /* exists */ and certificate
leaf[field.1.2.840.113635.100.6.1.13] /* exists */ and certificate leaf[subject.OU] =
"7AGZNQ2S2T"</string>
  <key>FilterPackets</key>
  <true/>
  <key>FilterSockets</key>
  <true/>
  <key>FilterType</key>
  <string>Plugin</string>
  <key>PayloadDisplayName</key>
  <string>Web Content Filter</string>
  <key>PayloadIdentifier</key>
  <string>com.apple.webcontent-filter.71C289AC-7ACF-44BC-AB5E-580736C634DF</string>
  <key>PayloadType</key>
  <string>com.apple.webcontent-filter</string>
  <key>PayloadUUID</key>
  <string>71C289AC-7ACF-44BC-AB5E-580736C634DF</string>
  <key>PayloadVersion</key>
  <integer>1</integer>
  <key>PluginBundleID</key>
  <string>com.vmware.carbonblack.cloud.se-agent</string>
  <key>UserDefinedName</key>
  <string>Carbon Black Network Extension Filter</string>
</dict>
```

- 4 Save the configuration profile.

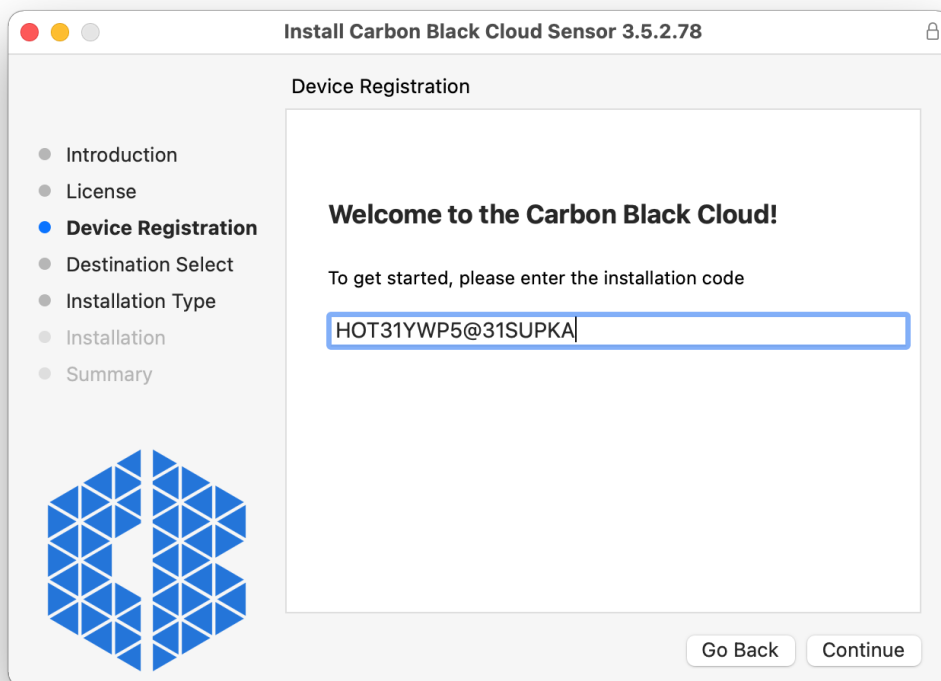
Manually Install and Approve the Sensor on macOS 11+

After publishing the device configuration profile, manually install and approve the Carbon Black Cloud sensor on macOS Big Sur (macOS 11+).

Procedure

- 1 Start the Carbon Black Cloud installer and click **OK** when the installer requests access to your Desktop folder.
- 2 Enter the sensor installation code.

If you enter the installation code incorrectly, an error message states that the installer cannot communicate with the Carbon Black Cloud. Check the installation code and try again.

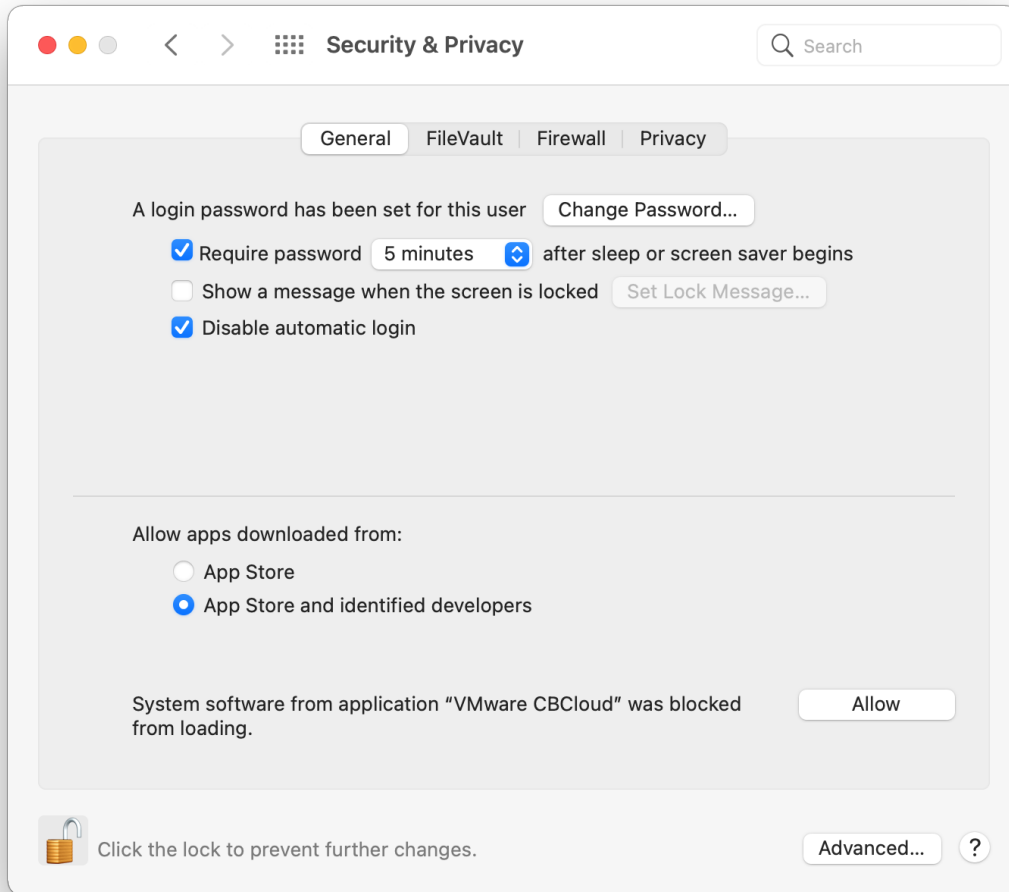


When the installer finishes running, a message notifies you that you must approve the VMware Carbon Black Cloud system extension.

- 3 Click **Open Security Preferences** to open the **Security & Privacy** pane.

On the **General** tab in the **Security & Privacy** pane, a notification indicates that the VMware CBCloud system extension was blocked from loading.

- 4 Enter your administrator password to unlock the pane and then click the **Allow** button next to the notification.



Note The notification persists, regardless of device restart, and the sensor remains in bypass mode until System Extensions approval.

After the System Extension is approved, another notification states that the VMware Carbon Black Cloud sensor wants to filter network content.

- 5 Click **Allow**.

Enabling Network Extensions is required for the sensor to report network events to the Carbon Black Cloud console.

Note If you click **Don't Allow** instead of **Allow**, network events do not reach the backend. You can restart this prompt by running the `CBCloud.app` in the `/Applications/VMware Carbon Black Cloud` folder.

Confirm the Carbon Black Cloud Sensor Installed on the macOS Device

You can use the RepCLI output to verify that the Carbon Black Cloud sensor for macOS is installed successfully on the device.

Procedure

- 1 Open the `Terminal.app` on the enrolled macOS device and enter the following command:
`cd /Applications/VMware\ Carbon\ Black\ Cloud/repcli.bundle/Contents/MacOS`
- 2 Then, enter the following command: `sudo ./repcli status`.

If prompted, enter the administrative password.

- 3 Observe the values for `System Extension status`, `sensor State`, and `Cloud Status` lines.

A successful deployment lists the values as follows.

| Option | Description |
|-------------------|-------------|
| System Extension: | Running |
| State: | Enabled |
| Cloud Status: | Registered |

What to do next

- If the System Extensions are not loading, make sure that you staged the correct profile payloads.
- If the Kernel Extensions are not loading in macOS Big Sur, you must rebuild the kernel cache.

Rebuild the Kernel Extension Cache

If you installed the Carbon Black Cloud sensor for macOS in Kernel Extension mode and the Kernel Extensions are not loading, you must rebuild the Kernel Extension cache.

Procedure

- 1 In the Workspace ONE UEM admin console, navigate to **Devices > List View** and select the macOS device that needs the kernel cache rebuilt.
- 2 Click the **More Actions** drop-down menu and select **Custom Command**.
- 3 Paste the following command in the **Command XML** text box and add the full list of `KextPaths` into the `array`.

```
<dict>
  <key>RebuildKernelCache</key>
  <true/>
  <key>KextPaths</key>
  <array>
    <string>/Library/Extensions/CbDefenseSensor.kext</string>
    <string>/Library/Extensions/SomeOtherExtension.kext</string>
```

```

    </array>
    <key>RequestType</key>
    <string>RestartDevice</string>
  </dict>

```

Note When you specify the `<key>KextPaths</key>`, you must include the Carbon Black Cloud Kernel Extension path, as well as any other paths you want to include in the Kernel Cache Rebuild.

- 4 Optional. Paste the following command in the **Command XML** text box without the `<key>KextPaths</key>` and array values, if they are unknown.

```

<dict>
  <key>RebuildKernelCache</key>
  <true/>
  <key>RequestType</key>
  <string>RestartDevice</string>
</dict>

```

Note If you do not specify the `<key>KextPaths</key>`, macOS attempts to rebuild the cache with any known Kernel Extensions. For example, from Apps that have launched before and attempted to load a Kernel Extension.

- 5 Click **Send**.

Deploying the Carbon Black Cloud sensor for macOS as Managed Application

As a Workspace ONE administrator, you can create a non-store, managed application for macOS in Workspace ONE. You must supply the icon file, the installer (dmg or pkg), and the metadata file. Additionally, the sensor kit deployment package structure requires modification to the metadata (PLIST) file. Then, you can deploy the Carbon Black Cloud sensor to an enrolled macOS device.

Prerequisites

- Use the Workspace ONE Intelligent Hub for macOS to determine, based on the metadata file, if the managed application is installed and if the installed application is the correct version.
- Become familiar with the Workspace ONE Intelligent Hub app. For details, see *Overview of VMware Workspace ONE Intelligent Hub*, which is part of *VMware Workspace ONE UEM Documentation*.
- Before configuring the sensor kit deployment, you must generate the required icon and metadata file with the Workspace ONE Admin Assistant application.

- Make sure you are familiar with the VMware Workspace ONE Admin Assistant tool. For more information, see *Workspace ONE Admin Assistant → Admin Assistant for macOS*, which is part of *VMware Workspace ONE Productivity Apps Documentation*.

Procedure

1 Set Up the Application Installer

To correctly distribute the Carbon Black Cloud sensor for macOS as a managed application, you must parse the sensor kit and modify the `PLIST` file.

2 Install the Carbon Black Cloud Sensor for macOS as a Managed Application

As a Workspace ONE administrator, after you modify the `PLIST` file you can deploy the Carbon Black Cloud sensor to an enrolled macOS device.

3 Confirm the Carbon Black Cloud Sensor for macOS Installed as Managed Application

There are few ways that you can validate if the Carbon Black Cloud sensor for macOS has been successfully installed as managed application on your macOS device.

Set Up the Application Installer

To correctly distribute the Carbon Black Cloud sensor for macOS as a managed application, you must parse the sensor kit and modify the `PLIST` file.

Procedure

- 1 Open the Workspace One Admin Assistant app and upload the downloaded sensor kit from the Carbon Black Cloud console (`confer_installer_mac-<version>.dmg`).
- 2 When parsing completes, click the **Reveal in Finder** button.
- 3 Expand the `CBCloud Install-<version>` folder and right-click the `CBCloud Install-<version>.plist` file.
- 4 Click **Open With** and select an editor of your choice.
For example, TextEdit, Xcode, or vim.
- 5 Modify the `PLIST` file by adding the following XML snippet.

```
<key>installs</key>
  <array>
    <dict>
      <key>CFBundleIdentifier</key>
      <string>com.vmware.carbonblack.cloud.se-agent</string>
      <key>CFBundleName</key>
      <string>VMware CBCloud</string>
      <key>CFBundleShortVersionString</key>
      <string>3.5.1fc19</string>
      <key>CFBundleVersion</key>
      <string>3.5.1fc19</string>
      <key>minosversion</key>
      <string>10.15</string>
      <key>path</key>
```

```

    <string>/Applications/VMware Carbon Black Cloud/VMware CBCloud.app</string>
    <key>type</key>
    <string>application</string>
    <key>version_comparison_key</key>
    <string>CFBundleShortVersionString</string>
  </dict>
</array>

```

- 6 Replace the `CFBundleShortVersionString` and `CFBundleVersion` values if they are different for the particular sensor version that you are deploying.
- 7 Optional. Generate the `installs` array in either way:
 - Export the `VMware CBCloud.app` from the installer package and run `VMware CBCloud.app` through the Workspace ONE Admin Assistant app.
 - If you installed the Carbon Black sensor kit on the machine with Workspace ONE Admin Assistant, copy `VMware CBCloud.app` to your `~/Downloads` directory (`cp -R /Applications/VMware\ Carbon\ Black\ Cloud\VMware CBCloud.app ~/Downloads`) and parse `~/Downloads/VMware CBCloud.app` through the Workspace ONE Admin Assistant app.

The PLIST generated in this instance contains the appropriate `installs` array information.

- 8 Save and close the modified PLIST in the editor of your choice.

Install the Carbon Black Cloud Sensor for macOS as a Managed Application

As a Workspace ONE administrator, after you modify the PLIST file you can deploy the Carbon Black Cloud sensor to an enrolled macOS device.

Add an Application

Use the Workspace ONE UEM admin console to deploy an internal macOS application for the sensor.

Prerequisites

To have the Workspace ONE UEM natively process macOS metadata, install the Workspace ONE UEM Admin Assistant for macOS tool. For description of the Admin Assistant tool and how to install it, see *Introduction to Workspace ONE Admin Assistant for macOS*, part of the *VMware Workspace ONE UEM* documentation.

Procedure

- 1 Select **Resources** from the **Getting Started** wizard.
- 2 Unfold **Apps** and select **Native > Internal**.
- 3 Select **Application File** from the **Add** drop-down menu.
- 4 Click **Upload**, choose the local file `confer_installer_mac-<version>.dmg` that you generated with the VMware Workspace ONE UEM Admin Assistant Tool, and save it.

- 5 Upload the metadata file by choosing the `CbDefense Install-<version>.plist` file that you generated with the Workspace ONE Admin Assistant, and save it.
- 6 Add an image for the app install by dragging the `CBCloud Install.png` graphic to the Workspace ONE UEM console.

Define Pre-Install and Uninstall Scripts

You can provide a pre-install and post-install scripts to populate a configuration file that is consumed by the Carbon Black Cloud sensor kit installation.

You must configure the **Scripts** settings to run the installation and uninstallation of the Carbon Black Cloud sensor macOS application. By providing pre-install scripts and post-install scripts, you can perform additional configuration tasks or install additional items without the need of repacking the application or software. Simply paste the script and Workspace ONE UEM formats it to be used by Munki.

Procedure

- 1 Select the **Scripts** tab.

- Paste one of the following scripts into the **Pre-Install Script** text box and replace the Code value with your Registration Code.

Each of the scripts includes the bare minimum required information — the Registration Code, for installing the Carbon Black Cloud sensor for macOS. The advanced pre-install script contains additional options for customizing the sensor installation. Replace them with your own values.

| Option | Description |
|--------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Basic Pre-Install Script For System Extension Install | <pre>#!/bin/bash PATH="/var/cbcloud-install" /bin/mkdir -p "\$PATH" /usr/bin/touch "\$PATH/cfg.ini" /bin/cat > "\$PATH/cfg.ini" <<- EOM [customer] Code=<REGISTRATION_CODE> DisableSysextnetworkExtension=false KernelType=2 EOM</pre> <p>For example:</p> <pre>#!/bin/bash PATH="/tmp/cbcloud-install" /bin/mkdir -p "\$PATH" /usr/bin/touch "\$PATH/cfg.ini" /bin/chmod 644 "\$PATH/cfg.ini" /bin/cat > "\$PATH/cfg.ini" <<- EOM [customer] Code=12345 DisableSysextnetworkExtension=false KernelType=2 EOM</pre> |
| Basic Pre-Install Script For Kernel Extension Install | <pre>#!/bin/bash PATH="/var/cbcloud-install" /bin/mkdir -p "\$PATH" /usr/bin/touch "\$PATH/cfg.ini" /bin/cat > "\$PATH/cfg.ini" <<- EOM [customer] Code=<REGISTRATION_CODE> DisableSysextnetworkExtension=false KernelType=1 EOM</pre> |
| Advanced Pre-Install Script | <pre>PATH="/var/cbcloud-install" /bin/mkdir -p "\$PATH" /usr/bin/touch "\$PATH/cfg.ini" /bin/cat > "\$PATH/cfg.ini" <<- EOM [customer] Code=<REGISTRATION_CODE> ProxyServer=<PROXY_SERVER> ProxyServerCredentials=<PROXY_CREDS> LastAttemptProxyServer=<LAST_ATTEMPT_PROXY_SERVER> PemFile=<customer.pem> AutoUpdate=<true false> AutoUpdateJitter=<true false> InstallBypass=<true false> FileUploadLimit=<FILE_UPLOAD_LIMIT></pre> |

| Option | Description |
|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | <pre> GroupName=<GROUP_NAME> EmailAddress=<USER_NAME> BackgroundScan=<true false> RateLimit=<RATE_LIMIT> ConnectionLimit=<CONNECTION_LIMIT> QueueSize=<QUEUE_SIZE> LearningMode=<LEARNING_MODE> <POC=1> CbLRKill=<true false> HideCommandLines=<true false> DisableSysextNetworkExtension=<true false> KernelType=<1 2> #1=KEXT,2=SysExt EOM </pre> |

- 3 Select **Uninstall Script** as the uninstall method.
- 4 Paste the script and populate the Deregistration Code into the **Uninstall Script** text box.

```

#!/bin/sh
/Applications/VMware\ Carbon\ Black\ Cloud/uninstall.bundle/Contents/MacOS/uninstall -y -c
<Deregistration_Code>

```

Set Deployment Options

By setting the deployment options you define the applications or processes that can prevent the installation from completing successfully.

Procedure

- 1 Click the **Deployment** tab.
- 2 Select **No** for the **Blocking Applications** option.
The end user must not close any Carbon Black Cloud applications. This is all handled by the Workspace ONE Intelligent Hub and the Carbon Black Cloud sensor installer.
- 3 If deploying the sensor with System Extensions, select **None** from the **Restart Action** drop-down menu. If deploying the sensor using KEXTs, choose the appropriate restart action.
- 4 Click **Save and Assign**.

Configure the Assignment

Before assigning the configurations to the sensor application installer, you must set your distribution and restriction preferences.

Procedure

- 1 In the **Distribution** page, enter a name for the distribution.
For example, *All Macs*.
- 2 Select the **Assignment Groups** containing the devices that must receive the Carbon Black Cloud sensor.

- 3 Select **Auto** for the **App Delivery Method** option.
- 4 Determine if you want the user to see the Carbon Black application in their App Catalog.
It can remain inactive.
- 5 Click **Restrictions** and enable **Remove on Unenroll** and **Desired State Management**.
- 6 **Create** the assignment.
- 7 Optional. Locate the **Exclusions** tab, add exclusions to the assignments, and adjust the priority for the assignments.
- 8 **Save** the assignment.
- 9 Review the assigned device and click **Publish**.

Confirm the Carbon Black Cloud Sensor for macOS Installed as Managed Application

There are few ways that you can validate if the Carbon Black Cloud sensor for macOS has been successfully installed as managed application on your macOS device.

Verify Carbon Black Cloud Sensor for macOS Installed as Managed Application with Workspace ONE UEM

You can use the Workspace ONE UEM admin console to verify that the Carbon Black Cloud sensor for macOS has installed as a managed application on the assigned devices.

Procedure

- 1 Go to **Devices > List View**
- 2 Select a device and click the **Apps** tab.
- 3 Locate the Carbon Black Cloud sensor for macOS in the list of applications.

Results

The Carbon Black Cloud sensor is installed as a managed application on the devices you previously assigned.

Verify Carbon Black Cloud Sensor for macOS Installed as Manage Application on the Device

You can check if the Carbon Black Cloud sensor for macOS application has been installed successfully on your macOS device.

Procedure

- 1 Open **Finder > Application** on the enrolled macOS device.
- 2 Ensure that the `VMware Carbon Black Cloud` folder is present and contains the **VMware CBCloud** sensor application and its related bundles.
- 3 Optionally check if the Confer menulet exists in the menu bar of the device.

Verify Carbon Black Cloud Sensor for macOS Installed as Managed Application in the Installation Log

You can check if the Carbon Black Cloud sensor for macOS application has been installed successfully on your macOS device by viewing the installation logs.

Procedure

- 1 Open the `Terminal.App` on the enrolled macOS device and enter the following command:

```
tail -f -n +1 /Library/Application\ Support/AirWatch/Data/Munki/Managed\ Installs/Logs/  
ManagedSoftwareUpdate.log
```

- 2 Browse the log file for the line stating that the `Install of CbDefense Install.pkg` was successful.

If the line states that the sensor appears to not be installing, or is installing repeatedly, you must adjust the metadata of the `PLIST` file to include an `installs` array.

Installing Sensors on Endpoints in a VDI Environment

4

This section describes how to install sensors through the command line or software distribution tools in a Virtual Desktop Infrastructure (VDI) environment.

Important Before you begin the processes described here, read [Chapter 1 Getting Started with Sensor Installation](#). It contains highly relevant information to help you succeed in your sensor installation.

Before you install sensors, perform the following steps:

[Obtain a Company Registration Code](#)

[Download Sensor Kits](#)

For firewall and proxy information, see [Chapter 12 Configuring Carbon Black Cloud Communications](#).

This chapter includes the following topics:

- [Creating Multiple Golden or Primary Images](#)
- [Carbon Black Windows Sensors with VMware Horizon Virtual Desktops](#)
- [Carbon Black Linux Sensors with VMware Horizon Virtual Desktops](#)
- [Carbon Black Windows Sensors with Citrix Virtual Desktops](#)
- [Carbon Black Windows Sensors with vSphere Clients](#)
- [Carbon Black Linux Sensors with vSphere Clients](#)

Creating Multiple Golden or Primary Images

This topic describes caveats and steps to follow when creating (cloning) multiple golden or primary images. It is pertinent to all VDIs.

To create multiple primary images, you must uninstall the sensor from the cloned primary image. Then, install a new sensor. The second primary image will receive a base device ID on its own, thus severing the dependency of the original primary image.

If you do not follow this recommendation and if the cloned primary image is deregistered because of inactivity and policy settings, sensors on the VDIs will be uninstalled and the VDIs will be unprotected.

Create Multiple Golden or Primary Images

This section provides general steps to follow when creating (cloning) multiple golden or primary images.

Procedure

- 1 Clone the golden or primary image.
- 2 Start the cloned image device.
- 3 Log in to the Carbon Black Cloud console.
- 4 Click **Inventory > Endpoints** or click **Inventory > VM Workload**.
- 5 Verify that the cloned image has a different DeviceID from the original image.

Note If the cloned image does not have a different DeviceID, run the `RepCLI reregister now` command on the cloned image and then verify that it has a new DeviceID.

- 6 Uninstall the sensor from the cloned image.

Note Do not uninstall the sensor if the cloned image device does not have a different DeviceID.

- 7 Install the sensor on the cloned image device.

Carbon Black Windows Sensors with VMware Horizon Virtual Desktops

This section describes how to deploy Carbon Black Cloud Windows 3.6+ sensors on Horizon full clones and Horizon instant clones 7.13, Horizon 2012 or later.

Before you install the Carbon Black Cloud Windows sensor, make sure that you are following standard Horizon best practices, including optimizing the golden image. The versions of the Carbon Black Cloud sensor and Horizon that you are using must be compatible and supported. See the [VMware Interoperability Matrix](#).

Review the following requirements and implement the recommended best practices. If the best practices included here are not the preferred method for deployment, an alternative configuration that uses the `OFFLINE_INSTALL` switch is supported. This is useful for organizations who want to create a golden image and clone it to offline computers. See [Windows Sensor Supported Commands](#).

See [Chapter 4 Installing Sensors on Endpoints in a VDI Environment](#) for preliminary instructions.

Carbon Black Windows Sensor Policy Setting Recommendations for Horizon Instant Clones

We recommend that the golden image be assigned a different policy from its clones. Use sensor groups to avoid the clones inheriting the golden image.

For more information about sensor groups, see the *VMware Carbon Black Cloud User Guide*

To get started, we recommend that you duplicate the Standard policy rules to the instant clone policy. We then recommend the following specific policy settings for Horizon instant clones.

General Tab

- **Name** – For easy identification, we recommend giving the policy a name that distinguishes the sensors as Instant Clones.
- **Description** – This policy is optimized for Horizon instant clones. Special considerations improve performance and provide a strong base of reputation, behavioral, and targeted prevention.
- **Target Value** – Medium

Sensor Tab

- **Display sensor message in system tray** - Enable this setting and add a message similar to this sample text: "Virtual Desktops Policy - Contact *someone@example.com* with any questions and concerns. Provide context regarding the issue and any available replication steps."

Prevention Tab - Permissions

- **Bypass rules (exclusions)** – Policy-level bypass rules help achieve stability in a VDI environment.

Each organization must understand the trade-offs between performance and security. VMware recommends the use of exclusions. Work with stakeholders to review risks and benefits (performance versus visibility) and apply the bypass rules as needed.

Carbon Black Cloud provides exclusions for supported methods as examples. Please review the applications that are installed in the VDI environment and apply any required bypass rules.

The following examples are based on public documentation for VMware solutions. Additional bypass rules might be needed.

VMware bypass rules best practices

```
**\Program Files\VMware\**,
**\SnapVolumesTemp**,
**\SVROOT**,
**\SoftwareDistribution\DataStore**,
**\System32\Spool\Printers**,
**\ProgramData\VMware\VDM\Logs**,
**\AppData\VMware\**
```

Prevention

Blocking and Isolation

Best practices recommend applying **Blocking and Isolation** rules to address specific attack surfaces.

Local Scan tab

- **On Access File Scan Mode** – Enabled
- **Allow Signature Updates** – Disabled

This setting is circumstantial. For short-lived clones, it is recommended to have **Allow Signature Updates** set to Disabled and have **On Access File Scan Mode** set to Enabled. The policy of the golden image would have both of these settings Enabled. This setup makes sure that clones can still perform AV scans using the signature packs that came from the golden image, without incurring the cost of updating the signature pack on each clone. If the clones are expected to be long-lived it is recommended to have both settings set to Enabled (to avoid the use of outdated signature packs).

Sensor tab

- **Run Background Scan** – Disabled. To optimize performance, it is recommended to complete a background scan on the golden image and subsequently have the background scan disabled on the policy that is assigned to the clones.
- **Scan files on network drives** – Disabled
- **Scan execute on network drives** – Enabled
- **Delay execute for Cloud scan** – Enabled. This critical setting serves as the sole point of reference for pre-execution reputation lookups. If it is disabled, endpoints must rely on **Application at Path** and **Deny List** rules for pre-execution prevention.
- **Hash MD5** – Disabled. The sensor always calculates the SHA-256.
- **Auto-deregister VDI clone sensors that have been inactive for** – Because instant clones are generally short-lived, it is recommended to Enable this setting to remove any instant clones that have been inactive for the specified duration.

Note VMware Carbon Black recommends setting an interval of at least 24 hours to ensure that sensors do not get de-registered during common maintenance windows from VMware Carbon Black or your environment.

Install the Carbon Black Windows Sensor on Horizon Instant Clones

To install a Carbon Black Cloud Windows sensor on Horizon instant clones, perform the following procedure. These instructions apply to both instant clone pools and instant clone farms.

Important Previous installation use of a post-synchronization script (batch file) is no longer necessary. If you are upgrading to Horizon 7.13+ from a previous Horizon version, you must remove the batch file that had previously been inserted into the golden image. Failure to remove the script will cause multiple re-registrations of the same device.

Do not run `repcli reregister now` or `repcli reregister onrestart` commands on the golden image. Either command turns the golden image into a clone, which might deregister the golden image if autoderegister is set and a time-out has occurred. Deregistration of the golden image results in clones being unable to reregister.

The instant clone agent now sets the following registry value to a unique GUID when `IT/replica/clone nga` customization begins. Each clone has a unique value:

```
Key: HKLM\Software\VMware, Inc.\ViewComposer\ga\AgentIntegration
Type: REG_SZ
Value: CustomizationStarted
```

Prerequisites

See [Carbon Black Windows Sensor Policy Setting Recommendations for Horizon Instant Clones](#) before installing the sensor.

Procedure

- 1 Create the golden image VM for the clone pool deployment. Perform required Windows updates and install the required [VMware Tools](#) and Horizon Agent.
- 2 Install the sensor on the golden image:
 - If you are using Horizon versions 7.13+ or 8.0+ and Carbon Black Cloud sensor 3.6+, no additional configuration is required. In this case, the sensor uses a Horizon Agent-provided registration key to perform reregistration on the clone:

```
msiexec.exe /q /i <Sensor Installer Path> /L*v msi.log COMPANY_CODE="XYZABC"
CLI_USERS=<UserGroupSid>POLICY_NAME="<NAME Virtual Policy>"
```


- If you are using a Horizon version Pre-7.13, 8.0 and Windows sensor 3.7 MR2+, add the "AUTO_REREGISTER_FOR_VDI_CLONES=3" install flag:

```
msiexec.exe /q /i <Sensor Installer Path> /L*v msi.log COMPANY_CODE="XYZABC"
CLI_USERS=<UserGroupSid> AUTO_REREGISTER_FOR_VDI_CLONES=3 POLICY_NAME="<NAME Virtual
Policy>"
```

Note <Sensor Installer Path>: Replace this value with the location of the sensor MSI file; for example, c:\tmp\installer_win-64-3.8.0.627.msi.

CLI_USERS= <UserGroupSid>: This parameter on the golden image enables RepCLI usage on the clones. The value is the Security Identifier (SID) of the user account/group that will run RepCLI commands on the clones.

Policy_NAME: Indicates the policy name that has the necessary exclusions and configurations to apply to the golden image. For Carbon Black Cloud sensors that are on versions prior to 3.8, use GROUP_NAME parameter instead.

See [Installing Windows Sensors on Endpoints](#) and [Windows Sensor Supported Commands](#). For more information about RepCLI, see [Managing Sensors by using RepCLI](#) in the *VMware Carbon Black Cloud User Guide*.

- 3 Optional (Recommended). Complete a background scan on the golden image to optimize clone performance.
 - a In the Carbon Black Cloud console, click **Enforce > Policies**, select the policy, and click the **Sensor** tab.
 - b Select the **Run background scan** option and select **Expedited** scanning.
 - c Click **Save**.
 - d You can track scan progress by running the `repcli status` command. The output will be similar to the following:

```
General Info:
  Sensor Version[3.7.0.1473 - Sep 29 2021 - 20:34:38]
  Local Scanner Version[ - ]
  Disk Filter Version[3.7.0.1473]
  CbShared[104365] Policy[1269] FileAnalysis[386] Proto[548]
  Sensor State[Enabled]
  Details[LiveResponse:NoSession, LiveResponse:NoKillSwitch, LiveResponse:Disabled,
  SvcStable]
  DeviceHash[31dbad895ab7161f1f53bed2f4e3fa49ac64de98935b03752b53a407f65d9ea2]
  DeviceID[26365289]
  VirtualGuestToHostCommsStatus[Disconnected]
  ExternalIdentity[Not Available]
  Kernel File Filter[Connected]
  LastUser[Device\user]
  Background Scan [Complete]
  Total Files Processed[52581] Current Directory[None]
```

4 Optional. Configure cache persistence for improved performance.

The persistent cache setting (`FileCachePersistenceState=3`) saves significant CPU and disk IO resources by reusing calculated hashes on clones. This feature is available with Windows sensor 3.8+. In addition, pruning parameters (available with Windows sensor 3.7+) improve VDI performance.

Persistent cache depends on the secure storage of the golden image snapshot files and assumes that no modifications are made to the snapshot while the golden image is offline. When enabling this setting, secure the golden image and storage infrastructure to an equivalent level, or to a higher level than the guest OS. We recommend that you limit physical and administrative access to the golden image and storage infrastructure, and regularly check your audit logs.

- a In the left navigation bar in the console, click **Inventory>Endpoints** or **Inventory>Workloads**.
- b Select the endpoint, click **Take Action**, and click **Enable bypass**. Confirm the action.
- c To confirm that the endpoint is in bypass mode, run the following command: `repcli status`.
- d As a best practice, make a backup of the `cfg.ini` file into another directory. For Windows sensor versions 3.6 and earlier, `cfg.ini` is located at `C:\Program Files\Confer\cfg.ini`. For Windows sensors 3.7+, `cfg.ini` is located at `C:\ProgramData\CarbonBlack\DataFiles\cfg.ini`.
- e Edit `cfg.ini`. Add the following parameters:
 - `RepDbPruneCountdownMs=14400000` (Default is 5 minutes; modified to 4 hours). This setting defines the interval after which the first pruning attempt is initiated.
 - `PruneDeletedFilesSleepInterval=14400000` (Default is 30 minutes; modified to hours). This setting defines the delay for subsequent attempts after a successful pruning attempt.
 - `PruneDeletedFilenameRowCount=500` (Default is 100). This setting defines the maximum number of rows to prune in one attempt.
 - `FileCachePersistenceState=3` (Default is 0). This setting enables cache persistence on instant clones.
- f Reboot the golden image.
- g Log into the golden image and run common applications to populate the cache.
- h In the left navigation bar in the console, click **Inventory>Endpoints** or **Inventory>Workloads**.
- i Select the endpoint, click **Take Action**, and click **Disable bypass**. Confirm the action.
- j Delete the backup file you created in Step 4d.
- k Shut down the golden image.

- 5 Reboot the golden image to apply full ransomware protections (3.7+ sensors). You can skip this step if you completed Step 4.
- 6 Take a snapshot of the golden image.
- 7 In the Horizon console, create an instant clone pool using the golden image and snapshot.
- 8 After the pool becomes available in the Horizon console, check in the Carbon Black Cloud console to verify that the newly created instant clones are registered with a new Device ID and are assigned the correct policy. The endpoint inherits the policy from the golden image unless you have previously created sensor groups and the installed sensor matches a sensor group's criteria. Manual policy assignment post-installation overrides the inheritance.

Horizon Instant Clones and Carbon Black Windows Sensor Installation Known Issues and Mitigation

During the instant clone pool creation, a temporary full clone of the golden image known as the “internal template” (with a device name `itxxxxxxx`) powers on and has network access. The sensor on that internal template device will probably connect to the Carbon Black Cloud with the same Device ID as the golden image. This connection results in the golden image being overwritten by the `itxxxxxxx` device in the Carbon Black Cloud console.

Note This issue is resolved with the Carbon Black Cloud Windows sensor 3.7MR2.

When the golden image is powered on, the sensor on the golden image re-connects to the backend and overwrites the `itxxxxxxx` device. In addition to the duplicate devices overwriting each other's data on the backend, this can lead to the backend sending a re-register request to the golden image. This causes the golden image to be considered a VDI by the backend, which could cause the golden image to deregister due to inactivity.

The duplicate device scenario can also expose a group membership issue where the golden image is no longer in the expected policy group. The negative implications of the internal template having the duplicate device ID as the golden image are as follows:

- The internal template's events and activities can intermingle with the golden images.
- The golden image's device name in the console might change.
- If you are using MSM to assign device policy by device name, make sure that golden image and internal template names are accounted for.

We recommend that you deploy instant clones with the golden image powered off. This recommendation will not eliminate the internal template duplicate Device ID scenario, but it will mitigate the downside of having a duplicate Device ID.

Provisioning of instant clones with the setting **Provision all machines up front** can cause a measurable increase in CPU utilization on the hosts and the provisioning clones during the initial provisioning operation. This is a result of the Carbon Black sensor registration occurring on all the provisioning instant clone at the same time.

Provisioning of instant clones with the setting **Provision machines on demand** is unlikely to cause an increase in CPU utilization because the instant clones are provisioned at different times.

As a result of higher CPU load, during initial instant clone provisioning the customization might time-out for the clones while executing the post-synchronization batch file. This can cause an error state for the clones. The error state is due to the 20 seconds timeout limit for executing the post synchronization scripts on the clones, but the failed clones will auto-recover. The 20 seconds default timeout of the Post Synchronization script can be adjusted by modifying the following registry key in the golden image. This reduces the instant clones provisioning failure rate. The maximum suggested value is 120000ms.

```
HKLM\System\CCS\Services\vmware-viewcomposer-ga
Type: DWORD
Value Name: ExecScriptTimeout
Units: milliseconds
Sample:
##Updated timeout value from 20000 to 120000 .
HKLM\System\CCS\Services\vmware-viewcomposer-ga
Type: DWORD
Value Name: ExecScriptTimeout
Units: 120000
```

Carbon Black Windows Sensor Policy Setting Recommendations for Horizon Full Clones

We recommend that the golden image be in a separate policy from its clones. Use sensor groups to avoid the clones inheriting the golden image policy.

For more information about sensor groups, see the *VMware Carbon Black Cloud User Guide*

To get started, we recommend that you duplicate the Standard policy rules to the full clone policy. We then recommend the following specific policy settings for Horizon full clones.

General Tab

- **Name** – For easy identification we recommend giving the policy a name that distinguishes the sensors as Full Clones.
- **Description** – This policy is optimized for Horizon full clones. Special considerations improve performance and provide a strong base of reputation, behavioral, and targeted prevention.
- **Target Value** – Medium

Sensor Tab

- **Display sensor message in system tray** - Enable this setting and add a message similar to this sample text: "Virtual Desktops Policy - Contact *someone@example.com* with any questions and concerns. Provide context regarding the issue and any available replication steps."

Prevention Tab - Permissions

- **Bypass rules (exclusions)** – Policy-level bypass rules help achieve stability in a VDI environment.

Each organization must understand the trade-offs between performance and security. VMware recommends the use of exclusions. Work with stakeholders to review risks and benefits (performance versus visibility) and apply the bypass rules as needed.

Carbon Black Cloud provides exclusions for supported methods as examples. Please review the applications that are installed in the VDI environment and apply any required bypass rules.

The following examples are based on public documentation for VMware solutions. Additional bypass rules might be needed.

VMware bypass rules best practices

```
**\Program Files\VMware\**,
**\SnapVolumesTemp**,
**\SVROOT**,
**\SoftwareDistribution\DataStore**,
**\System32\Spool\Printers**,
**\ProgramData\VMware\VDM\Logs**,
**\AppData\VMware\**
```

Prevention

Blocking and Isolation

Best practices recommend applying **Blocking and Isolation** rules to address specific attack surfaces.

Local Scan tab

- **On Access File Scan Mode** – Enabled
- **Allow Signature Updates** – Enabled

Full clones are rarely recreated from the golden image, so they effectively never receive signature updates. Enable **Allow Signature Updates** for full clones.

Sensor tab

- **Run Background Scan** – Disabled. To optimize performance, it is recommended to complete a background scan on the golden image and then subsequently have the background scan disabled on the policy assigned to the clones.
- **Scan files on network drives** – Disabled
- **Scan execute on network drives** – Enabled

- **Delay execute for Cloud scan** – Enabled. This critical setting serves as the sole point of reference for pre-execution reputation lookups. If it is disabled, endpoints must rely on **Application at Path** and **Deny List** rules for pre-execution prevention.
- **Hash MD5** – Disabled. The sensor always calculates the SHA-256.
- **Auto-deregister VDI sensors that have been inactive for** – Disable this setting to prevent unintentional uninstall of the sensor.

Install Carbon Black Windows Sensors on Horizon Full Clones

If you are installing Carbon Black Cloud Windows sensors into an environment that is comprised of Horizon full clones only, use the following installation method.

Important Previous installation use of a post-synchronization script (batch file) is no longer necessary. If you are upgrading to Horizon 7.13+ from a previous Horizon version, you must remove the batch file that had previously been inserted into the golden image. Failure to remove the script will cause multiple re-registrations of the same device.

Do not run `repcli reregister now` or `repcli reregister onrestart` commands on the golden image. Either command turns the golden image into a clone, which might deregister the golden image if autoderegister is set and a time-out has occurred. Deregistration of the golden image results in clones being unable to reregister.

Prerequisites

See [Carbon Black Windows Sensor Policy Setting Recommendations for Horizon Full Clones](#) before installing the sensor.

Procedure

- 1 Create the golden image VM for the clone pool deployment. Perform required Windows updates and install the required [VMware Tools](#) and Horizon Agent.
- 2 Install the sensor on the golden image:
 - If you are using Horizon versions 7.13+ or 8.0+ and Carbon Black Cloud sensor 3.6+, no additional configuration is required. In this case, the sensor uses a Horizon Agent-provided registration key to perform reregistration on the clone:

```
msiexec.exe /q /i <Sensor Installer Path> /L*v msi.log COMPANY_CODE="XYZABC"
CLI_USERS=<UserGroupSid>POLICY_NAME="<NAME Virtual Policy>"
```

Note The instant clone agent now sets the following registry value to a unique GUID when IT/replica/clone nga customization begins. Each clone has a unique value:

```
Key: HKLM\Software\VMware, Inc.\ViewComposer\ga\AgentIntegration
Type: REG_SZ
Value: CustomizationStarted
```

- If you are using a Horizon version Pre-7.13, 8.0 and Windows sensor 3.7 MR2+, add the "AUTO_REREGISTER_FOR_VDI_CLONES=3" install flag:

```
msiexec.exe /q /i <Sensor Installer Path> /L*v msi.log COMPANY_CODE="XYZABC"
CLI_USERS=<UserGroupSid> AUTO_REREGISTER_FOR_VDI_CLONES=3 POLICY_NAME="<NAME Virtual
Policy>"
```

Note <Sensor Installer Path>: Replace this value with the location of the sensor MSI file; for example, c:\tmp\installer_win-64-3.8.0.627.msi.

CLI_USERS= <UserGroupSid>: This parameter on the golden image enables RepCLI usage on the clones. The value is the Security Identifier (SID) of the user account/group that will run RepCLI commands on the clones.

Policy_NAME: Indicates the policy name that has the necessary exclusions and configurations to apply to the golden image. For Carbon Black Cloud sensors that are on versions prior to 3.8, use GROUP_NAME parameter instead.

See [Installing Windows Sensors on Endpoints](#) and [Windows Sensor Supported Commands](#). For more information about RepCLI, see [Managing Sensors by using RepCLI](#) in the *VMware Carbon Black Cloud User Guide*.

- 3 Optional (Recommended). Complete a background scan on the golden image to optimize clone performance.
 - a In the Carbon Black Cloud console, click **Enforce > Policies**, select the policy, and click the **Sensor** tab.
 - b Select the **Run background scan** option and select **Expedited** scanning.
 - c Click **Save**.
 - d You can track scan progress by running the `repcli status` command. The output will be similar to the following:

```
General Info:
  Sensor Version[3.7.0.1473 - Sep 29 2021 - 20:34:38]
  Local Scanner Version[ - ]
  Disk Filter Version[3.7.0.1473]
  CbShared[104365] Policy[1269] FileAnalysis[386] Proto[548]
  Sensor State[Enabled]
  Details[LiveResponse:NoSession, LiveResponse:NoKillSwitch, LiveResponse:Disabled,
SvcStable]
  DeviceHash[31dbad895ab7161f1f53bed2f4e3fa49ac64de98935b03752b53a407f65d9ea2]
  DeviceID[26365289]
  VirtualGuestToHostCommsStatus[Disconnected]
  ExternalIdentity[Not Available]
  Kernel File Filter[Connected]
  LastUser[Device\user]
  Background Scan [Complete]
  Total Files Processed[52581] Current Directory[None]
```

- 4 Optional: Update the device signature of the golden image by running the `repcli updateAVSignature` command.
- 5 Reboot the golden image to apply full ransomware protections (Windows sensor versions 3.7+).
- 6 Shut down the golden image. In the Horizon console, convert the golden image into the template VM. Create a full clone pool using the golden VMTemplate.
- 7 New full clones will register with a new Device ID in the Carbon Black Cloud console after the pool becomes available. Confirm that newly provisioned clones have registered and are assigned the correct policy.

Install Carbon Black Windows Sensors in Horizon Full and Instant Clone Mixed Environments

Use the following procedure to install Carbon Black Cloud Windows sensors in an environment that has both full and instant Horizon clones.

Procedure

- 1 Create the golden image for the full clone pool. As per the Horizon documentation, perform the required steps, including installing [VMware Tools](#) and Horizon Agent. Do not install the Carbon Black Cloud sensor on the golden image.
- 2 Put the Carbon Black Cloud sensor MSI on the golden image (preferably in the System Root directory).
- 3 Prepare the Customization Specification that will be used to create the full clone pool.
- 4 Add the following sensor installation command into Customization Specification commands:

```
msiexec.exe /q /i <Sensor Installer Path> /L*v msi.log COMPANY_CODE="XYZABC"
CLI_USERS=<UserGroupSid> GROUP_NAME="<NAME Virtual Policy>"
```

Note For Horizon Pre-7.13, 8.0 and Windows sensors 3.7MR2+, add the following parameter to enable automatic reregistration of clones: `AUTO_REREGISTER_FOR_VDI_CLONES=3`. If you are using an older sensor version, use the `BASE_IMAGE=1` parameter instead.

< Sensor Installer Path> : Replace this value with the location of the sensor MSI file; for example, `c:\tmp\installer_win-64-3.6.0.1941.msi`.

`CLI_USERS=` *UserGroupSid*: This parameter enables RepCLI usage on the clones. The value is the Security Identifier (SID) of the user account/group that will run RepCLI commands on the clones.

`GROUP_NAME`: Indicates the policy name that has the necessary exclusions and configurations to apply to the clones.

See [Chapter 5 Installing Windows Sensors on Endpoints](#) and [Windows Sensor Supported Commands](#). For more information about RepCLI, see [Managing Sensors by using RepCLI](#) in the user guide.

- 5 Deploy the full clone pool from the golden image VMTemplate by using the Customization Specification.

The cloned VMs are registered to the Carbon Black Cloud console. If enabled by policy, a background scan is run on each cloned VM after the pool is provisioned. Note that the background scan can cause performance issues, depending on how many VMs exist per host.

- 6 Enable sensor settings to deregister inactive VMs. This setting provides operational and management benefits to instant clone VMs.
 - a In the Carbon Black Cloud console, go to **Inventory>Endpoints>Sensor Options>Sensor Settings** or **Inventory>Workloads>Sensor Options>Sensor Settings** or **Inventory>VDI Clones>Sensor Options>Sensor Settings**.
 - b For instant clones, enable the following options and set the timeframes to ensure automatic clean-up of inactive, deregistered instant clones. Do not enable these options for full clones.
 - **Delete sensors that have been deregistered for...**
 - **Deregister VDI sensors that have been inactive for...**
- 7 Enable the **Signature Update** setting on **Enforce>Policy>Assigned Policy>Local Scan>Signature Update**.

Note This installation method requires a different golden image for the full clone pool than for the instant clone pool.

With the 3.7+ Windows sensor, a reboot is needed on VDI clones to fully apply new ransomware protections.

Horizon Linked-Clones and Carbon Black 3.6+ Windows Sensor Best Practices

Carbon Black supports Horizon linked-clones; however, linked-clones are scheduled for extended support and end-of-life. Linked-clones are deprecated in Horizon 8. Carbon Black recommends migrating to Horizon version 7.13 and later versions, with the Carbon Black Cloud Windows 3.6 sensor or later.

See [Horizon Linked-Clones](#).

You can manage the performance impact of the Carbon Black Cloud sensor with linked-clones by ensuring the following:

- 1 The correct permission bypass (exclusions) are in place at the Policy level.
- 2 A background scan is completed on the golden image VM prior to using vCenter Server to take a snapshot of the golden image VM.

A background scan takes several hours to complete. There is a significant performance benefit from running it on the golden image. Completing the background scan enables the sensor to gather cloud reputation for hashes found on the golden image. This removes the need for the linked-clones to delay execution to pull reputation when those hashes eventually run.

Carbon Black Linux Sensors with VMware Horizon Virtual Desktops

This section describes how to deploy Carbon Black Cloud Linux sensors on Horizon virtual desktops.

See [Chapter 4 Installing Sensors on Endpoints in a VDI Environment](#) for preliminary instructions.

Before you install Linux sensors in a Horizon VDI environment, confirm that your environment meets the minimum requirements.

- Carbon Black Cloud sensors: Linux sensor v2.12 and later
- Horizon 8.1 and later
- OS distributions and versions must be supported by the Linux sensor and Horizon

Carbon Black Linux Sensor Policy Setting Recommendations for Horizon Golden Images

We recommend that Carbon Black Cloud console administrators create specific policies to manage a Horizon golden image.

After a policy is applied to the golden image, all clones inherit this policy unless otherwise directed by membership in sensor groups.

For more information about sensor groups, see the *VMware Carbon Black Cloud User Guide*.

We recommend the following policy settings for a Horizon golden image.

- Duplicate the Standard policy and make the following changes on the **Sensor** tab.
- **Run Background Scan** – For optimal clone performance, run the background scan on the golden image. This pre-populates the sensor cache with the reputation of files that are currently on the system and improves clone performance. A background scan takes some time to complete, and not all users want to wait for the scan when creating a new image. For performance sensitive customers, the extra wait time might be worth it if the image is deployed at scale. Turn this setting OFF after the background scan is complete.

- **Auto-deregister VDI clone sensors that have been inactive for** – Enable this setting to remove any instant clones that been inactive for the specified duration. Set the timeframe to remove inactive VMs.

Note VMware Carbon Black recommends setting an interval of at least 24 hours to ensure that sensors do not get de-registered during common maintenance windows from VMware Carbon Black or your environment.

Note Previously, golden image sensors could be inadvertently uninstalled by auto-deregistration settings. This is no longer possible because the backend will not deregister any device that is the golden image for a clone.

Horizon Golden Image Considerations for Carbon Black Linux Sensors

This article contains recommendations for installing the Carbon Black Linux sensor on the Horizon golden image.

- Make sure that the golden image never registers as a clone or gets deregistered.
- For optimal performance, allow the background scan to complete on the golden image before creating clones.

Important Do not run “`/opt/carbonblack/psc/bin/cbagentd -R`” on the golden image. This command converts the golden image into a clone, which might de-register the golden image if auto-deregister of VDI clone sensors is set and timeout has reached. The deregister of golden image results in failure of clone registration.

Horizon Instant Clone Considerations for Carbon Black Linux Sensors

This article describes how a Horizon instant clone receives a Device ID and is assigned to a policy.

- 1 The endpoint requests a new Device ID.
- 2 The new Device ID is identified as a virtual desktop on the backend.
- 3 The endpoint inherits the policy from the golden image unless you have previously created sensor groups and the installed sensor matches a sensor group’s criteria. Manual policy assignment post-installation overrides the inheritance.

Install the Carbon Black Linux Sensor on a Horizon Golden Image and Create Instant Clones

Use the following procedure to install the Carbon Black Linux sensor on a Horizon golden image and create instant clones.

Prerequisites

See the following topics:

- [Chapter 4 Installing Sensors on Endpoints in a VDI Environment](#)
- [Carbon Black Linux Sensors with VMware Horizon Virtual Desktops](#)

Procedure

- 1 Create the golden image for the clone pool deployment. Perform required Linux updates, dependency installation for Horizon, and Horizon agent installation.
- 2 Install the Linux sensor on the golden image by following the steps in [Installing Linux Sensors on Endpoints](#).
- 3 Confirm that configuration properties are set to enable the automatic identification and registration of a Horizon instant clone:
 - a The features are enabled by default. If `EnableAutoReregisterForVDIClones` does not exist in `/var/opt/carbonblack/psc/cfg.ini`, then you do not need to do anything and can proceed directly to Step 4. If the following property exists in the configuration file, confirm that it is set to one of the following values to enable automatic registration:
 - `EnableAutoReregisterForVDIClones=3`. Enables BIOS UUID and MAC Address hash change-based automatic registration. Recommended and default setting.
 - `EnableAutoReregisterForVDIClones=2`. Enables BIOS UUID change-based automatic registration.
 - b If the `EnableAutoReregisterForVDIClones` property is not set to the correct value, perform the following steps:
 - 1 Run `service cbagentd stop` or `systemctl stop cbagentd`.
 - 2 Edit the `/var/opt/carbonblack/psc/cfg.ini` file to set the correct value.
 - 3 Run `service cbagentd start` or `systemctl start cbagentd`.

Note To disable automatic registration, set the value of `EnableAutoReregisterForVDIClones` to 1.

- 4 Allow the background scan to complete on the golden image to optimize clone performance. Run the following command to determine whether the background scan has completed:

```
cat /var/opt/carbonblack/psc/blades/E51C4A7E-2D41-4F57-99BC-6AA907CA3B40/th.ini | grep LocalScanRunning
```

If `LocalScanRunning` is true, the background scan is ongoing.

- 5 Power off the golden image.

- 6 Take a snapshot of the golden image.

Important Do not run the `/opt/carbonblack/psc/bin/cbagentd -R` command on the golden image. This command turns the golden image into a clone, which might deregister pre-existing clones.

- 7 In the Horizon console, create an instant clone pool using the golden image and the snapshot created in Step 6.
- 8 After the pool becomes available in the Horizon console, verify that the newly created instant clones are registered with a new Device ID in the Carbon Black Cloud console.

Carbon Black Windows Sensors with Citrix Virtual Desktops

This section describes how to deploy Carbon Black Cloud Windows sensors on Citrix virtual desktops.

Note Linux sensors are not supported on Citrix virtual desktops.

Citrix Golden Image Considerations for Carbon Black Sensors

This topic contains recommendations for installing the Carbon Black Cloud sensor on a Citrix golden image.

- For optimal clone performance, allow the background scan to complete on the golden image before creating clones.
- You can run the `repcli status` command to view the background scan status. For example:

```
General Info:
  Sensor Version[3.7.0.1473 - Sep 29 2021 - 20:34:38]
  Local Scanner Version[ - ]
  Disk Filter Version[3.7.0.1473]
  CbShared[104365] Policy[1269] FileAnalysis[386] Proto[548]
  Sensor State[Enabled]
  Details[LiveResponse:NoSession, LiveResponse:NoKillSwitch, LiveResponse:Disabled,
  SvcStable]
  DeviceHash[31dbad895ab7161f1f53bed2f4e3fa49ac64de98935b03752b53a407f65d9ea2]
  DeviceID[26365289]
  VirtualGuestToHostCommsStatus[Disconnected]
  ExternalIdentity[Not Available]
```

```
Kernel File Filter[Connected]
LastUser[Device\user]
Background Scan [Complete]
Total Files Processed[52581] Current Directory[None]
```

Important Do not run `repcli reregister now` or `repcli reregister onrestart` commands on the golden image. Either command turns the golden image into a clone, which might deregister the golden image if `autoderegister` is set and a time-out has occurred. Deregistration of the golden image results in clones being unable to reregister.

Carbon Black Policy Setting Recommendations for Citrix Golden Images

We recommend that Carbon Black Cloud console administrators create specific policies to manage a Citrix golden image.

All clones inherit the policy from the golden image unless otherwise directed by membership in a sensor group.

For more information about sensor groups, see *VMware Carbon Black Cloud User Guide*.

We recommend the following policy settings for a Citrix golden image.

Prevention Tab - Permissions

- **Bypass rules (exclusions)** – Policy-level bypass rules help achieve stability in a VDI environment.

Each organization must understand the trade-offs between performance and security. VMware recommends the use of exclusions. Work with stakeholders to review risks and benefits (performance versus visibility) and apply the bypass rules as needed.

Carbon Black Cloud provides exclusions for supported methods as examples. Please review the applications that are installed in the VDI environment and apply any required bypass rules.

The following examples are based on public documentation for Citrix solutions. Additional bypass rules might be needed.

Citrix bypass rules best practices

```
**\Program Files*\Citrix\**,
  **\AppData\Local\Temp\Citrix\HDXRTConnector\*\*.txt,
  **\*.vdiskcache,
  **\System32\spoolsv.exe
```

Note Additional bypass rules might be required. For example, some organizations do not want to bypass `winlogon.exe`. This is a Citrix recommendation for any AV solution because a common problem with VDIs that use AV is longer login times. This bypass rule helps restore the expected experience.

Prevention

Blocking and Isolation

Best practices recommend applying **Blocking and Isolation** rules to address specific attack surfaces. To get started, we recommend that you duplicate the Standard policy rules to the Virtual Desktops policy.

Local Scan tab

- **On Access File Scan Mode** – Disabled
- **Allow Signature Updates** – Enabled

Sensor tab

- **Run Background Scan** – For optimal clone performance, run the background scan on the golden image. This pre-populates the sensor cache with the reputation of files that are currently on the system and improves clone performance. A background scan takes some time to complete, and not all users want to wait for the scan when creating a new image. For performance sensitive customers, the extra wait time might be worth it if the image is deployed at scale.
- **Scan files on network drives** – Disabled
- **Scan execute on network drives** – Enabled
- **Delay execute for Cloud scan** – Enabled. This critical setting serves as the sole point of reference for pre-execution reputation lookups. If it is disabled, endpoints must rely on **Application at Path** and **Deny List** rules for pre-execution prevention.
- **Hash MD5** – Disabled. The sensor always calculates the SHA-256.
- **Auto-deregister VDI sensors that have been inactive for** – Disable this setting to prevent unintentional uninstall of the sensor.

Note Previously, Carbon Black Cloud could automatically deregister golden image machines due to inactivity. Carbon Black Cloud no longer leverages time-based deregistration for any VM that has a child.

Install the Sensor on a Citrix Golden Image

Use the following procedure to set up the Carbon Black Cloud sensor on a Citrix golden image virtual machine (VM).

Procedure

- 1 Create the golden image VM.

2 Install the sensor on the golden image using the following command:

```
msiexec.exe /q /i <Sensor Installer Path> /L*v msi.log COMPANY_CODE="XYZABC"
CLI_USERS=<UserGroupSid> GROUP_NAME="<NAME Virtual Policy>"
```

Note For Windows sensors 3.7MR1+, add the `AUTO_REREGISTER_FOR_CITRIX=true` parameter to the command line.

< Sensor Installer Path > : Replace this value with the location of the sensor MSI file; for example, `c:\tmp\installer_win-64-3.6.0.1941.msi`.

`CLI_USERS=UserGroupSid`: This parameter on the golden image enables RepCLI usage on the clones. The value is the Security Identifier (SID) of the user account/group that will run the `reregister now` command on the clone.

`GROUP_NAME`: Indicates the policy name that has the necessary exclusions and configurations to apply to the golden image.

See [Chapter 5 Installing Windows Sensors on Endpoints](#) and [Windows Sensor Supported Commands](#). For more information about RepCLI, see [Managing Sensors by using RepCLI](#) in the User Guide.

3 Complete an expedited background scan on the golden image to optimize clone performance.

- a In the Carbon Black Cloud console, click **Enforce > Policies**, select the policy, and click the **Sensor** tab.
- b Select the **Run background scan** option and select **Expedited** scanning.
- c Click **Save**.
- d You can track scan progress by running the `repcli status` command. The output will be similar to the following:

```
General Info:
  Sensor Version[3.7.0.1473 - Sep 29 2021 - 20:34:38]
  Local Scanner Version[ - ]
  Disk Filter Version[3.7.0.1473]
  CbShared[104365] Policy[1269] FileAnalysis[386] Proto[548]
  Sensor State[Enabled]
  Details[LiveResponse:NoSession, LiveResponse:NoKillSwitch, LiveResponse:Disabled,
  SvcStable]
  DeviceHash[31dbad895ab7161f1f53bed2f4e3fa49ac64de98935b03752b53a407f65d9ea2]
  DeviceID[26365289]
  VirtualGuestToHostCommsStatus[Disconnected]
  ExternalIdentity[Not Available]
  Kernel File Filter[Connected]
  LastUser[Device\user]
  Background Scan [Complete]
  Total Files Processed[52581] Current Directory[None]
```

4 Apply the clone policy to the golden image. For recommendations on clone policy settings, see [Carbon Black Policy Setting Recommendations for Citrix Clones](#).

- 5 Take a snapshot of the golden image.

Note Previously, the Carbon Black Cloud could automatically deregister golden image machines due to inactivity. The Carbon Black Cloud no longer leverages time-based deregistration for any VM that has a child.

Citrix Clone Considerations for Carbon Black Windows Sensors

This topic describes how a Citrix clone receives a Device ID and is assigned to a policy.

The `reregister` command is needed to register new clones with a unique Device ID.

When `reregister now` is run, a clone performs the following operations:

- 1 The endpoint requests a new Device ID.
- 2 The new Device ID is identified as a VDI endpoint on the backend.
- 3 The endpoint inherits the policy from the primary image unless you have previously created sensor groups and the installed sensor matches a sensor group's criteria. Manual policy assignment post-installation overrides the inheritance.

Note With the 3.7+ Windows sensor, a reboot is needed on VDI clones to fully apply new ransomware protections.

Carbon Black Policy Setting Recommendations for Citrix Clones

We recommend that Carbon Black Cloud console administrators create specific policies to manage Citrix clones.

After a policy is applied to the golden image, all clones inherit this policy unless otherwise directed by membership in sensor groups.

For more information about sensor groups and policy settings, see the *VMware Carbon Black Cloud User Guide*.

We recommend the following policy settings for Citrix clones.

General Tab

- **Name** – Virtual Desktops – “Virtual Desktops” was previously a prescribed policy name. You can now put VMs into any policy name, and support VMs in different policies. This allows you to segregate clones from physical machines, and have different settings for each type.
- **Description** – This policy is optimized for Citrix clones. Special considerations improve performance and provide a strong base of reputation, behavioral, and targeted prevention.
- **Target Value** – Medium

Sensor Tab

- **Display sensor message in system tray** - Enable this setting and add a message similar to this sample text: "Virtual Desktops Policy - Contact someone@example.com with any questions and concerns. Provide context regarding the issue and any available replication steps."

Prevention Tab - Permissions

- **Bypass rules (exclusions)** – Policy-level bypass rules help achieve stability in a VDI environment.

Each organization must understand the trade-offs between performance and security. VMware recommends the use of exclusions. Work with stakeholders to review risks and benefits (performance versus visibility) and apply the bypass rules as needed.

Carbon Black Cloud provides exclusions for supported methods as examples. Please review the applications that are installed in the VDI environment and apply any required bypass rules.

The following examples are based on public documentation for Citrix solutions. Additional bypass rules might be needed.

Note Additional bypass rules might be required. For example, some organizations do not want to bypass `winlogon.exe`. This is a Citrix recommendation for any AV solution because a common problem with VDIs that use AV is longer login times. This bypass rule helps restore the expected experience.

Citrix bypass rules best practices

```
**\Program Files*\Citrix\**,
    **\AppData\Local\Temp\Citrix\HDXRTConnector\*\*.txt,
    **\*.vdiskcache,
    **\System32\spoolsv.exe
```

Prevention

Blocking and Isolation

Best practices recommend applying **Blocking and Isolation** rules to address specific attack surfaces. To get started, we recommend that you duplicate the Standard policy rules to the Virtual Desktops policy.

Local Scan tab

- **On Access File Scan Mode** – Disabled
- **Allow Signature Updates** – Disabled

It is a best practice to disable **Allow Signature Updates** for clones. The local scan feature adds network overhead and augments resource utilization. The Carbon Black Cloud can pull reputation and enforce policy in real time from the Cloud because most VDI environments maintain 99% uptime.

However, you can install the signature pack to the golden image. This installation avoids the performance penalty of running updates on each clone, but allows the clones to have some offline protection. Malware that can be identified by the signature pack on the golden image is detected and blocked independent of Cloud activity.

Installing updates to a golden image works well for clones because the clones are frequently recreated from the golden image and thereby inherit the updates.

Sensor tab

- **Run Background Scan** – To optimize performance, most VDI vendors recommend disabling any background scan of the file system. Operating under the expectation that the golden image is free of malware, and the clones maintain consistent connectivity to the Cloud, it is not recommended to utilize the background scan feature. Reputation is derived from the Cloud at execution when necessary, per policy configuration. See the following **Delay Execute for Cloud** scan recommendation.
- **Scan files on network drives** – Disabled
- **Scan execute on network drives** – Enabled
- **Delay execute for Cloud scan** – Enabled. This critical setting serves as the sole point of reference for pre-execution reputation lookups. If it is disabled, endpoints must rely on **Application at Path** and **Deny List** rules for pre-execution prevention.
- **Hash MD5** – Disabled. The sensor always calculates the SHA-256.
- **Auto-deregister VDI sensors that have been inactive for** – Enable this setting to remove any clones that been inactive for the specified duration.

Citrix MCS and Carbon Black Windows 3.7MR1+ Sensor

This topic describes the necessary steps to deploy Citrix MCS with the Carbon Black Cloud Windows 3.7MR1+ sensor.

Procedure

- 1 Run the following command to install the Windows 3.7MR1+ sensor:

```
msiexec.exe /q /i <Sensor Installer Path> /L*v msi.log COMPANY_CODE="XYZABC"
CLI_USERS=<sid> GROUP_NAME="<NAME Virtual Policy>" AUTO_REREGISTER_FOR_CITRIX=true
```

< Sensor Installer Path > : Replace this value with the location of the sensor MSI file; for example, c:\tmp\installer_win-64-3.6.0.1941.msi.

CLI_USERS= *sid*: This parameter enables RepCLI usage. The value is the Security Identifier (SID) of the user account/group that will run RepCLI commands .

GROUP_NAME: Indicates the policy name to apply.

See [Chapter 5 Installing Windows Sensors on Endpoints](#) and [Windows Sensor Supported Commands](#). For more information about RepCLI, see [Managing Sensors by using RepCLI](#) in the User Guide.

- 2 Check policy recommendations at [Carbon Black Policy Setting Recommendations for Citrix Clones](#).
- 3 Shut down the Golden Device.
- 4 Take a snapshot of the Golden Device.
- 5 Create a Citrix Machine Catalog based on the Golden Device snapshot.
- 6 Create a Citrix Delivery Group based on the newly created Citrix Machine Catalog.
- 7 Verify that Citrix clone devices are visible in the Carbon Black Cloud console.

Important Do not perform manual reregistration on the Golden Device.

Remove any previous BAT scripts or other reregister mechanisms that you have set up. Leaving such mechanisms in place can cause sensors to reregister more than once.

Citrix PVS and Carbon Black Windows 3.7MR1+ Sensor

This topic describes the necessary steps to deploy Citrix PVS with the Carbon Black Cloud Windows 3.7MR1+ sensor.

Procedure

- 1 Put the vDisk into Private mode and access vDisk from the *Designated Parent Device*.
- 2 Run the following command to install the Windows 3.7MR1+ sensor:

```
msiexec.exe /q /i <Sensor Installer Path> /L*v msi.log COMPANY_CODE="XYZABC"
CLI_USERS=<UserGroupSid> GROUP_NAME="<NAME Virtual Policy>" AUTO_REREGISTER_FOR_CITRIX=true
```

< *Sensor Installer Path* > : Replace this value with the location of the sensor MSI file; for example, c:\tmp\installer_win-64-3.6.0.1941.msi.

CLI_USERS= *UserGroupSid*: This parameter enables RepCLI usage. The value is the Security Identifier (SID) of the user account/group that will run RepCLI commands .

GROUP_NAME: Indicates the policy name to apply.

See [Chapter 5 Installing Windows Sensors on Endpoints](#) and [Windows Sensor Supported Commands](#). For more information about RepCLI, see [Managing Sensors by using RepCLI](#) in the User Guide.

- 3 Shut down the Designated Parent Device.
- 4 Put the vDisk into Shared mode.
- 5 Start Citrix clone devices with vDisk in Shared mode.

- 6 Verify that Citrix clone devices are visible in the Carbon Black Cloud console.

Important Do not use clone devices to access vDisk in Private mode. Always use vDisk in Private mode or Maintenance mode from the Designated Parent Device. Do not perform manual reregistration on the Parent Device.

Remove any previous BAT scripts or other reregister mechanisms that you have set up. Leaving such mechanisms in place can cause sensors to reregister more than once.

Citrix PVS and Carbon Black Windows 3.7 Sensor

This topic describes the necessary steps to deploy Citrix PVS with the Carbon Black Cloud Windows 3.7 sensor. Earlier sensor versions are not supported.

Procedure

- 1 Put the vDisk into Private mode and access vDisk from the *Designated Parent Device*.
- 2 Install or upgrade to the Windows 3.7 sensor.
- 3 In the Carbon Black Cloud console, click **Inventory>Endpoints** or **Inventory>Workloads**.
- 4 Select the endpoint, click **Take Action**, and then click **Enable bypass**. Confirm the action.
- 5 Edit `C:\ProgramData\CarbonBlack\DataFiles\cfg.ini` and append the following statement. A separate script to re-register the agent is not required after specifying this parameter in the `cfg.ini` file.

```
AutoReRegisterForCitrix=true
```

Note If you are upgrading to the 3.7 sensor from a previous sensor version, add:
`HostNameAsOfLastReregister=<HOSTNAME>`. Replace HOSTNAME with the hostname of the Designated Parent Device.

- 6 Apply the new configuration by using the following RepCLI command. For more information about RepCLI, see [Managing Sensors by using RepCLI](#) in the User Guide.

```
RepCLI updateconfig
```

- 7 In the Carbon Black Cloud console, click **Inventory>Endpoints** or **Inventory>Workloads**.
- 8 Select the endpoint, click **Take Action**, and then click **Disable bypass**. Confirm the action.
- 9 Shut down the Designated Parent Device.
- 10 Put the vDisk into Shared mode.
- 11 Start Citrix clone devices with vDisk in Shared mode.

- 12 Verify that Citrix clone devices are visible in the Carbon Black Cloud console.

Important Do not use clone devices to access vDisk in Private mode. Always use vDisk in Private mode or Maintenance mode from the Designated Parent Device. Do not perform manual reregistration on the Parent Device.

Remove any previous BAT scripts or other reregister mechanisms that you have set up. Leaving such mechanisms in place can cause sensors to reregister more than once.

Carbon Black Windows Sensors with vSphere Clients

This article describes how to install Carbon Black Cloud Windows sensors on the command line or through software distribution tools in a VMware vSphere environment to enable the automatic identification and registration of vSphere clones.

Prerequisites

To get started, see [Chapter 4 Installing Sensors on Endpoints in a VDI Environment](#).

Make sure that your environment meets the minimum requirements:

- Carbon Black Cloud Windows sensor 3.7 MR2 and later
- vCenter Server 6.7 or later
- Host with ESXi 6.7 or later connected to the vCenter server
- Windows VM running OS versions that are supported by the sensor
- Carbon Black Host Module installed and running on the hosts on which VMs are deployed

Procedure

- 1 Install the Windows sensor on VMs using the following command:

```
msiexec.exe /q /i <Sensor Installer Path> /L*v msi.log COMPANY_CODE="XYZABC"
CLI_USERS=<UserGroupSid> AUTO_REREGISTER_FOR_VDI_CLONES=3 GROUP_NAME="<Virtual Policy>"
```

< Sensor Installer Path > : Replace this value with the location of the sensor MSI file; for example, `c:\tmp\installer_win-64-3.7.0.1503.msi`.

`CLI_USERS= <UserGroupSid>`: This parameter enables RepCLI usage on the clones. The value is the Security Identifier (SID) of the user account/group that will run RepCLI commands on the clones.

`GROUP_NAME`: Indicates the policy name that has the necessary exclusions and configurations.

See [Chapter 5 Installing Windows Sensors on Endpoints](#) and [Windows Sensor Supported Commands](#). For more information about RepCLI, see [Managing Sensors by using RepCLI](#) in the User Guide.

2 (Optional) Confirm that the required configuration properties are set correctly to enable the automatic identification and reregistration of vSphere clones:

- a Use the `repcli configprops` command to verify that the configuration properties have the expected values:

Table 4-1. Configuration Properties

| Configuration Property and Value | Usage |
|--------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>VHostEnabled=1</code> | Controls communication with Host User World. Supported values: 0=False 1=True (default) |
| <code>EnableAutoReregisterFor VDIclones=3</code> | Controls the auto-reregistration feature. Supported values: 1=Disable auto-reregistration for VDI clones (default) 2=Make reregister decision based on BIOS UUID change only 3=Make reregister decision based on BIOS UUID and MAC HASH change (required) |
| <code>EnableExternalIdsChange DetectionForVDIclones=1</code> | Controls Host User World-based auto-reregistration feature. Supported values: 0=False 1=True (default) |
| <code>IncludeExternalIds InMsgsToBackend=1</code> | Select if external identifiers should be sent in messages to the backend. Supported values: 0=False 1=True (default) |

- b If any of these configuration properties have different values, edit `C:\ProgramData\CarbonBlack\DataFiles\cfg.ini` to change them. To disable automatic reregistration, set `EnableExternalIdsChangeDetectionForVDIclones` to 0. After you have made your changes, run the `RepCLI updateconfig` command to immediately update the `cfg.ini` file.

Note You must put the sensor into bypass mode before you can edit `cfg.ini`. As a best practice, make a backup of `cfg.ini` into another directory before you edit it in a plain text editor. After you have edited `cfg.ini`, take the sensor out of bypass mode. For more details about editing `cfg.ini`, see [How To Change ConfigProps Via Cfgi.ini](#).

- 3 Issue the `repcli status` command to verify that the sensor can connect to the host module and can query external identifiers. For example:

```
cmd> repcli status
General Info:
  Sensor Version[3.7.0.1500 - Oct  8 2021 - 14:41:39]
  Local Scanner Version[ - ]
  Sensor State[Enabled]
  DeviceHash[e56223a76e00...]
  DeviceID[11223344]
  VirtualGuestToHostCommsStatus[Connected]
  ExternalIdentity[ee8fc1c7-
bd1e-4b10-9f32-04825a8b136e::501052cb-1d88-24b6-963d-9625a6c39f1e]
```

- 4 Clone the VM through vSphere or by using APIs (managed object browser, pyVmomi, etc.).
The sensor running on the cloned VM should reregister. You can check the same by running `repcli status` on the clone VM and ensuring that the updated external identifiers have persisted and the Device ID has changed.

Carbon Black Linux Sensors with vSphere Clients

This article describes how to install Carbon Black Cloud Linux sensors through the command line or software distribution tools in a vSphere environment to enable the automatic identification and registration of VC clones.

Prerequisites

To get started, see [Chapter 4 Installing Sensors on Endpoints in a VDI Environment](#).

Make sure that your environment meets the minimum requirements:

- Carbon Black Cloud Linux sensor v2.12 and later
- Host with ESXi 6.7 or later
- vCenter Server 6.7 or later
- OS distributions and version must be supported by the Linux sensor

Procedure

- 1 Follow the steps described in [Enable Host User World](#) to install the HostUW module on the ESXi host.
- 2 Install the Linux sensor on the primary VM. See [Chapter 2 Installing Linux Sensors on Endpoints](#).

- 3 Confirm that configuration properties are set to enable the automatic identification and registration of a vSphere clone. For more information about `cfg.ini`, see [About the Linux Sensor `cfg.ini` File](#).
 - a The features are enabled by default. If the following properties do not exist in `/var/opt/carbonblack/psc/cfg.ini`, then you do not need to do anything and can proceed directly to Step 4. If the following properties exist in the configuration file, confirm that they are set to the following correct values to enable automatic registration.
 - `VHostEnabled=true`. Enables the automatic registration feature.
 - `EnableExternalIdsChangeDetectionForVDIClones=true`. Enables HostUW-based automatic registration.
 - `IncludeExternalIdsInMsgsToBackend=true`. Enables external identifiers in messages sent to the Cloud backend.

Note To optionally *disable* automatic registration, set the value of `EnableExternalIdsChangeDetectionForVDIClones=false`.

 - b If any property is not set to the correct value, perform the following steps:
 - 1 Run `service cbagentd stop` OR `systemctl stop cbagentd`
 - 2 Edit the `/var/opt/carbonblack/psc/cfg.ini` file to set the correct values.
 - 3 Run `service cbagentd start` OR `systemctl start cbagentd`.
- 4 Clone the primary VM.

Installing Windows Sensors on Endpoints

5

This section describes how to install Carbon Black Cloud Windows sensors on the command line or through software distribution tools.

Important Before you begin the processes described here, read [Chapter 1 Getting Started with Sensor Installation](#). It contains highly relevant information to help you succeed in your sensor installation.

Before you can install sensors, perform the following steps:

[Obtain a Company Registration Code](#)

[Download Sensor Kits](#)

You can optionally verify the Windows sensor signatures before installing the sensor. See [Verifying Windows Sensor Digital Signatures](#).

If you are installing Windows sensors v3.5 or later, you can install sensors offline. This is useful for organizations who want to create a primary image and clone it to offline computers. This option is only available if you are installing sensors on the command line, or by using software distribution tools. See also [Windows Sensor Supported Commands](#).

Note With the release of the Windows 3.6 sensor, you can supply either the installation code (obtained via email — see [Invite Users to install Sensors](#)) or the company code (obtained via the console — see [Obtain a Company Registration Code](#)).

For VMware Workspace One installation instructions, see [Deploying VMware Carbon Black Cloud Sensor with Workspace ONE UEM](#).

Important The 3.6 Windows sensor leverages a content management system to enable dynamic configuration of prevention features. Prior to installing or updating to 3.6, if you have restrictive firewall policies active in your environment, you might need to add a new firewall/proxy exclusion for the sensor to be fully functional. See [Configure a Firewall](#).

This chapter includes the following topics:

- [Verifying Windows Sensor Digital Signatures](#)
- [Windows Sensor Rollback](#)
- [Local Scan Settings and the AV Signature Pack](#)

- [Windows Sensor Command Line Parameters](#)
- [Windows Sensor Supported Commands](#)
- [Windows Command Line Install on Endpoints — Examples](#)
- [Windows Sensor Log Files and Installed Services](#)
- [Installing Windows Sensors on Endpoints by using Group Policy](#)
- [Installing Windows Sensors on Endpoints by using SCCM](#)
- [Installing Carbon Black Cloud Sensor for Windows by Using Workspace ONE UEM](#)

Verifying Windows Sensor Digital Signatures

You can optionally verify digital signatures of Windows sensor installation files.

Prepare to Verify Windows Sensor Digital Signatures

Perform the following steps to prepare to verify Windows sensor digital signatures.

Procedure

- 1 Download the [Microsoft Windows SDK](#).
- 2 Install all components of the SDK.

Note SignTool is usually installed under `C:\Program Files (x86)\Windows Kits\10\bin`, but the exact location depends on the version of the SDK and your operating system. For example, it can be installed in any of the following (or other) locations:

- `C:\Program Files (x86)\Windows Kits\10\App Certification Kit\signtool.exe`
 - `C:\Program Files (x86)\Windows Kits\10\bin\x86\signtool.exe`
 - `C:\Program Files (x86)\Windows Kits\10\bin\x64\signtool.exe`
-

- 3 Add the location of the `Signtool` binary to your PATH environment variable.
 - a Press the **Windows** key.
 - b Type **env**.
 - c Click **Edit the System Environment Variables**.
 - d Click **Environmental Variables**.
 - e Select **Path** and click **Edit**.

- f At the end of the existing value, add the `Signtool` location. A semicolon (;) must separate the old value from the new value. For example:
 - `old value = %USERPROFILE%\AppData\Local\Microsoft\WindowsApps;`
 - `new value = %USERPROFILE%\AppData\Local\Microsoft\WindowsApps;C:\Program Files (x86)\Windows Kits\10\App Certification Kit\`
- g Click **OK** three times to save your changes and exit the editor.

Verify the Signature of a Windows Sensor Install Package

Run the following procedure to verify the signature of a Windows sensor install package.

Procedure

- 1 Open a command prompt window.
- 2 Run the following command, where *\$file_to_verify* is the name of the install package:

```
signtool.exe verify /pa /hash SHA56 /all $file_to_verify
```

- The `/pa` parameter instructs `Signtool` to check for code signing.
- An optional `/hash SHA256` parameter instructs `Signtool` to only check the SHA256 signatures.
- The `/all` parameter instructs `Signtool` to check all signatures on the file.

Verify Multiple Files Digital Signatures

You can follow this procedure to verify multiple Windows sensor files. This procedure generally applies to installed products/packages.

Prerequisites

You must know which files can be verified. Typically, files that cannot be verified change during the use of the product, such as configuration files or JIT compiled files.

Procedure

- 1 Create a file that contains a list of files to verify, one file name per line. The following example includes relevant files for a x64 install package:

```
C:\users\user_name\desktop\cbd-setup64-3.8.0.276.msi
C:\program files\confer\api-ms-win-core-console-l1-1-0.dll
C:\program files\confer\api-ms-win-core-datetime-l1-1-0.dll
C:\program files\confer\api-ms-win-core-debug-l1-1-0.dll
C:\program files\confer\api-ms-win-core-errorhandling-l1-1-0.dll
C:\program files\confer\api-ms-win-core-file-l1-1-0.dll
C:\program files\confer\api-ms-win-core-file-l1-2-0.dll
C:\program files\confer\api-ms-win-core-file-l2-1-0.dll
C:\program files\confer\api-ms-win-core-handle-l1-1-0.dll
```

```

C:\program files\confer\api-ms-win-core-heap-l1-1-0.dll
C:\program files\confer\api-ms-win-core-interlocked-l1-1-0.dll
C:\program files\confer\api-ms-win-core-libraryloader-l1-1-0.dll
C:\program files\confer\api-ms-win-core-localization-l1-2-0.dll
C:\program files\confer\api-ms-win-core-memory-l1-1-0.dll
C:\program files\confer\api-ms-win-core-namedpipe-l1-1-0.dll
C:\program files\confer\api-ms-win-core-processenvironment-l1-1-0.dll
C:\program files\confer\api-ms-win-core-processthreads-l1-1-0.dll
C:\program files\confer\api-ms-win-core-processthreads-l1-1-1.dll
C:\program files\confer\api-ms-win-core-profile-l1-1-0.dll
C:\program files\confer\api-ms-win-core-rtlsupport-l1-1-0.dll
C:\program files\confer\api-ms-win-core-string-l1-1-0.dll
C:\program files\confer\api-ms-win-core-synch-l1-1-0.dll
C:\program files\confer\api-ms-win-core-synch-l1-2-0.dll
C:\program files\confer\api-ms-win-core-sysinfo-l1-1-0.dll
C:\program files\confer\api-ms-win-core-timezone-l1-1-0.dll
C:\program files\confer\api-ms-win-core-util-l1-1-0.dll
C:\program files\confer\api-ms-win-crt-conio-l1-1-0.dll
C:\program files\confer\api-ms-win-crt-convert-l1-1-0.dll
C:\program files\confer\api-ms-win-crt-environment-l1-1-0.dll
C:\program files\confer\api-ms-win-crt-filestream-l1-1-0.dll
C:\program files\confer\api-ms-win-crt-filesystem-l1-1-0.dll
C:\program files\confer\api-ms-win-crt-heap-l1-1-0.dll
C:\program files\confer\api-ms-win-crt-locale-l1-1-0.dll
C:\program files\confer\api-ms-win-crt-math-l1-1-0.dll
C:\program files\confer\api-ms-win-crt-multibyte-l1-1-0.dll
C:\program files\confer\api-ms-win-crt-private-l1-1-0.dll
C:\program files\confer\api-ms-win-crt-process-l1-1-0.dll
C:\program files\confer\api-ms-win-crt-runtime-l1-1-0.dll
C:\program files\confer\api-ms-win-crt-stdio-l1-1-0.dll
C:\program files\confer\api-ms-win-crt-string-l1-1-0.dll
C:\program files\confer\api-ms-win-crt-time-l1-1-0.dll
C:\program files\confer\api-ms-win-crt-utility-l1-1-0.dll
C:\program files\confer\concrtdll.dll
C:\program files\confer\msvcpl140.dll
C:\program files\confer\ucrtdll.dll
C:\program files\confer\vccorlib140.dll
C:\program files\confer\vcruntime140.dll
C:\program files\confer\BladeRunner.exe
C:\program files\confer\CbNativeMessagingHost.exe
C:\program files\confer\RepCLI.exe
C:\program files\confer\RepMgr.exe
C:\program files\confer\RepUtils.exe
C:\program files\confer\RepUx.exe
C:\program files\confer\RepWAV.exe
C:\program files\confer\RepWmiUtils.exe
C:\program files\confer\RepWSC.exe
C:\program files\confer\Uninstall.exe
C:\program files\confer\VHostComms.exe
C:\program files\confer\blades\livequery\osqueryi.exe
C:\program files\confer\blades\livequery\exts\cbc_plugin_extension.ext.exe
C:\program files\confer\blades\livequery\exts\cbosqext.dll
C:\program files\confer\scanner\apcfile.dll
C:\program files\confer\scanner\apchash.dll
C:\program files\confer\scanner\avupdate.dll
C:\program files\confer\scanner\msvcr120.dll

```

```
C:\program files\confer\scanner\savapi.dll
C:\program files\confer\scanner\scew.dll
C:\program files\confer\scanner\scanhost.exe
C:\program files\confer\scanner\upd.exe
```

2 Create a batch file that contains the following text:

```
@echo off
set FILE=list_of_files
set numFiles=0
set numGoodSigs=0
setlocal ENABLEDELAYEDEXPANSION

for /f "delims== tokens=1,2" %%G in (%FILE%) do (
    if not exist "%%G\*" (
        set /a numFiles=numFiles+1
        (signtool verify /all /hash SHA256 /pa "%%G") && (set /a numGoodSigs=numGoodSigs+1)
    )

    @echo. & @echo.
)

set /a numBadSigs=numFiles-numGoodSigs

echo %numFiles% files checked
echo %numGoodSigs% verified files
echo %numBadSigs% UNverifiable files
```

3 Change the value of *FILE* in the batch file to specify the file that you created in Step 1.

4 Run the batch file.

Results

A summary of how many files could or could not be verified is written at the end of the output. For example:

```
File: C:\program files\confer\api-ms-win-core-console-l1-1-0.dll
Index  Algorithm  Timestamp
=====
0      sha1      Authenticode
1      sha256     RFC3161

Successfully verified: C:\program files\confer\api-ms-win-core-console-l1-1-0.dll

67 files checked
67 verified files
0 UNverifiable files
```

Windows Sensor Rollback

With the Carbon Black Cloud Windows 3.6 sensor and later, if a failure occurs during an initial install or uninstall, the endpoint will be returned to the state it was in prior to the attempt.

If a failure occurs during initial installation of the sensor, the sensor will rollback any changes made to the system. This includes files, services, and registry artifacts that were removed, thereby leaving the system in a clean state to reattempt installation. This rollback does not harm other services or files on the endpoint.

If a failure occurs during the uninstall of a sensor, the sensor will roll back any changes including files, services, and registry artifacts. This rollback does not harm other services or files on the endpoint. The endpoint continues to check into the console and is controlled through its designated policy.

With the Carbon Black Cloud Windows 3.7 sensor and later, rollback is supported for upgrade scenarios. If a failure occurs during sensor upgrades, the endpoint will be returned to the state it was in prior to the upgrade attempt by rolling back any changes including files, services, and registry artifacts. This rollback does not harm other services or files on the endpoint. The endpoint continues to check into the console and is controlled through its designated policy.

The following table describes types of rollbacks that different sensor versions support:

Table 5-1. Rollbacks supported by sensor versions

| | 3.5 and earlier | 3.6 | 3.7 and later |
|---------------|-----------------|---------------|---------------|
| Fresh install | Not supported | Supported | Supported |
| Uninstall | Not supported | Supported | Supported |
| Upgrade | Not supported | Not supported | Supported |

The following table describes the expected final sensor state when failures occur while upgrading from different sensor versions:

Table 5-2. Sensor states

| Upgrading from | Upgrading to | Upgrade failure point | System state after upgrade | Additional explanation |
|----------------|--------------|-----------------------|---------------------------------------|----------------------------------------|
| 3.5 | 3.7 or above | Uninstall of 3.5 | System could be left with partial 3.5 | 3.5 did not support uninstall rollback |
| 3.5 | 3.7 or above | Install of 3.7 | System left with no sensor | 3.5 did not support upgrade rollback |
| 3.6 | 3.7 or above | Uninstall of 3.6 | System left with 3.6 | 3.6 supports uninstall rollback |
| 3.6 | 3.7 or above | Install of 3.7 | System left with no sensor | 3.6 did not support upgrade rollback |
| 3.7 or above | 3.7 or above | Uninstalling old | System left with older version | Uninstall rollback supported |
| 3.7 or above | 3.7 or above | Installing new | System left with older version | Upgrade rollback supported |

Local Scan Settings and the AV Signature Pack

The AV Signature Pack is not packaged with the sensor installation, but should be downloaded and installed automatically after sensor installation based on policy settings. As a best practice, we recommend that you download and install the AV Signature Pack 10 seconds or more after sensor installation.

Note The local scan feature is only available for Windows sensors 2.0 and later.

The AV Signature Pack requires approximately 120MB at rest. During run time, 400MB is required because a second copy is created; the scan continues to function while signatures are being updated. After the update is complete, the old signatures are deleted. At least 200MB of memory is required to run the local scan.

Signature file updates are ON by default via a policy setting. You might encounter high bandwidth utilization upon sensor installation due to the initial signature file download. Subsequent updates following the initial install of the AV Signature Pack are differential. Therefore, setting a regular update schedule ensures that every subsequent update remains small.

To avoid network saturation during sensor installation, we recommend the following best practices:

- Install sensors in small batches.
- Set up a local mirror server for signature updates and configure your policy so that sensors download updates from the local server. See [Signature Mirror Instructions](#).
- Disable automatic signature updates. Deploy the initial signature pack by using the standalone installer, and then re-enable automatic signature updates.

To Disable Automatic Signature Updates and use the Standalone Installer

Use the following procedure to disable automatic signature updates.

Procedure

- 1 Sign in to the Carbon Black Cloud console.
- 2 Click **Enforce**, click **Policies**, and select the policy.
- 3 Click the **Local Scan** tab and disable **Signature Updates**. Click **Save**.
- 4 Install the sensors. Make sure the sensors are assigned a policy that has signature updates disabled (steps 1 and 2). Wait at least 10 seconds before you run the signature pack installer.
- 5 Click **Endpoints**, click **Sensor Options**, and click **Download sensor kits**.
- 6 Download the AV Signature Pack.

- 7 Run the following command under a user account that has full administrator rights by using system management software:

```
CbDefenseSig-YYYYMMDD.exe /silent
```

Note You can run the installation command through Live Response.

- 8 On the **Local Scan** tab on the Policies page, enable **Signature Updates**. After you save the changes to the selected policy, sensors in that policy begin to download the AV Signature Pack from Carbon Black Cloud servers in the next 5-60 minutes.

By default, updates download every 4 hours with a staggered update window of 4 hours. You can change these settings on the **Local Scan** tab of the Policies page.

To Update the AV Signature Pack by using the RepCLI Command

You can update the AV signature pack by using the RepCLI command.

Procedure

- 1 Log into the machine with a user account that matches the AD User or Group SID that was configured at the time of sensor install.
- 2 Open a command prompt window with administrative privileges.
- 3 Change the directory to `C:\Program Files\Confer`.
- 4 Type the following command: `repcli UpdateAvSignature`

Results

If the command is successful, the message **The request of AV signature update has been accepted** displays on the command line.

Note Active Directory-based SID authentication is not required to run the `repcli UpdateAVSignature` command. For more information about RepCLI, see [Managing Sensors by using RepCLI](#) in the User Guide.

Windows Sensor Command Line Parameters

The following command line parameters are used during a Windows command line sensor installation.

| Parameter | Required or Optional | Description |
|-----------|----------------------|---------------------------------------------------------------------------------------------|
| /q | Required | If you install without using this parameter, the user is prompted for an installation code. |
| /i | Required | This parameter tells the MSI to install. |

| Parameter | Required or Optional | Description |
|-----------|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| /L* | Optional | Creates an MSI install log file. |
| /L*vx | Optional | Creates a verbose MSI install log file. This is recommended over the /L* parameter because it provides more information to troubleshoot installation problems. |

Windows Sensor Supported Commands

You can use the following command line parameters during a Windows sensor install.

Using commands or command line parameters other than those listed in the following table can cause the installation to fail.

Carbon Black Cloud supports automatic detection of proxy settings; however, it does not prompt for or pass the machine's credentials for use in proxy authentication, if enabled. If proxy authentication is required for your environment, use the command line options to specify `PROXY_SERVER=value`, `PROXY_USER=value`, and `PROXY_PASSWD=value`. See [Configure a Proxy](#).

Note

- Command options are case-sensitive.
- With Windows sensors 3.3.0 and later, you can use the `RepCLI` command line tool to locally administer the sensor. For more information about `RepCLI`, see [Managing Sensors by using RepCLI](#) in the user guide.

| Command Options | Values | Supported with Updates | Notes |
|-----------------------------------------------|------------|------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>AUTO_CONFIG_MEM_DUMP=value</code> | true/false | Yes | Provides the ability to stop the sensor from automatically configuring the memory dump settings in the registry when set to false. Available for Windows sensors 3.5+. |
| <code>AUTO_REREGISTER_FOR_CITRIX=value</code> | true/false | Yes | Default is false. When set to true, this setting enables auto-reregistration for Citrix PVS and MCS clones. Available for Windows sensors 3.7MR1+. |

| Command Options | Values | Supported with Updates | Notes |
|--------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| AUTO_REREGISTER_FOR_VDI_CLONES=value | <p>4 - Checks for Hostname change (available from 3.8+)</p> <p>3 - Checks for BIOS UUID + MAC HASH changes (preferred)</p> <p>2 - Checks for BIOS UUID change</p> <p>1 - Auto Reregister disabled</p> | Yes | Default for Windows sensor 3.7MR2 is 1. Default for Windows sensor 3.8+ is 3. Sets auto-reregistration functionality for Horizon and vSphere VDI clones. Available for Windows sensors 3.7MR2+. |
| AUTO_UPDATE=value | 1/0 or true/false | No | Default is true (enable auto update). This setting toggles whether the sensor will accept backend-pushed upgrade requests. Turning this off will prevent the update from being pushed from the backend. |
| BACKGROUND_SCAN=value | 1/0 or true/false | No | Default is true. This setting toggles whether sensor will do an inventory of what hashes exist on the machine. Not applicable to Audit and Remediation Standalone. |
| BASE_IMAGE=value | 1/0 or true/false | No | Default is false; the installed image is a base image that can be cloned to child images. This option is not supported for Non-Persistent VDI installations, but is currently used for Persistent VDI installations. |
| BYPASS=value | 1/0 or true/false | No | <p>Default is false; setting it to true will enable bypass mode</p> <p>In bypass mode, the sensor does not send any data to the cloud: the sensor functions in a passive manner and does not interfere with or monitor the applications on the endpoint.</p> <p>Installing the sensor in bypass mode enables thorough testing for interoperability issues.</p> |
| CBLR_KILL=value | 1/0 | No | <p>A value of 1 disables Live Response functionality for the sensor. The default value is 0.</p> <p>Note Not reversible without reinstalling the sensor.</p> |

| Command Options | Values | Supported with Updates | Notes |
|---------------------------|-------------------------------------------------|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CLI_USERS=sid | SID value for authenticated users group | No | Use this field to enable the RepCLI tool. Any member in the specified user group can use the authenticated RepCLI commands. |
| COMPANY_CODE=value | Company registration code | No | Required for command line installations. |
| CONNECT_LIMIT=value | Number of connections per hour | No | Optional; default is no limit. |
| CURL_CRL_CHECK | 1/0 | Yes | Default is 1 (enabled). See Disable CURL CRL CHECK |
| DELAY_SIG_DOWNLOAD=value | 1/0 | No | Default is delay signature/definition download. We recommend that you do not change the default value. |
| FILE_UPLOAD_LIMIT=value | 4-byte integer representing number of megabytes | No | Example: value of 3 is a limit of 3*1024*1024 bytes; default value is 5. |
| GROUP_NAME=value | String value | No | Optional policy name assignment. Enclose this value with double quotes if the policy name includes spaces. Use this parameter for Windows sensors 3.7 and earlier. For Windows sensors 3.8+, use the POLICY_NAME parameter instead. |
| HIDE_COMMAND_LINES | 1/0 | No | Obfuscates command line inputs. Default is 0. |
| LAST_ATTEMPT_PROXY_SERVER | Value example: 10.101.100.99:8080 | No | Optional. Sensor will attempt cloud access by using this setting when all other methods fail (including dynamic proxy detection). |

| Command Options | Values | Supported with Updates | Notes |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| LEARNING_MODE=value | <p>Value is the number of hours after sensor install to limit event types.</p> <p>This is a mechanism for reducing the load on the backend by dropping some report types after initial install.</p> <p>Generally, more reports are sent to the backend soon after sensor install because the sensor reports on newly detected hashes.</p> <p>Learning mode reports only on file and process behavior while the sensor is detecting hashes. Reporting of API, registry, and network behavior is dropped during this period.</p> | No | Optional; default is disabled. |
| OFFLINE_INSTALL=value | 1/0 or true/false | No | Optional. Default is false. This parameter allows you to install sensors when the endpoint is offline. The sensor connects with the Carbon Black Cloud backend and accesses a policy when network connectivity is restored. The sensor is in a bypass state until the sensor can access the policy. This option is only available for Windows sensors v3.5 and later. |
| POLICY_NAME=value | String value | No | Optional policy name assignment. Enclose this value with double quotes if the policy name includes spaces. Use this parameter for Windows sensors 3.8+. For Windows sensors 3.7 and earlier, use the GROUP_NAME parameter instead. |
| PROXY_PASSWD=value | Proxy password | No | Optional. |
| PROXY_SERVER=value | server:port | No | Optional. |
| PROXY_USER=value | Proxy username | No | Optional. |

| Command Options | Values | Supported with Updates | Notes |
|-------------------|-----------------------------------------------|------------------------|---------------------------------------------------------------------------------------------------|
| QUEUE_SIZE=value | Event backlog | No | Optional; default value for Endpoint Standard is 100MB; this value does not include SSL overhead. |
| RATE_LIMIT=value | KB per hour | No | Optional; default is No Limit. |
| USER_EMAIL=value | Email address Example: user@example.com | No | Optional. |
| VDI=value | 1/0 or true/false | No | This option is deprecated in sensor versions 3.4+. Default is false. |
| VHOST_COMMS=value | true/false | Yes | Disables the VHostComms helper utility when set to false. |

Obfuscation of Command Line Inputs

Endpoint users might input sensitive data into the command line. The obfuscation of command line inputs protects against unauthorized users accessing the data in plain text in the sensor `.log` files and the sensor databases.

There are three ways to obfuscate command line inputs:

- **Command line** - `HIDE_COMMAND_LINES=1`
- **RepCLI** - `hideCmdLines [0|1]`
- **Set** `HideCommandLines=true` in `cfg.ini`

The setting enables the obfuscation of command line input in sensor `.log` files and databases. The data in the Carbon Black Cloud console is not obfuscated. For more information about RepCLI, see [Managing Sensors by using RepCLI](#) in the User Guide.

Windows Command Line Install on Endpoints — Examples

Review the following examples of Windows sensor installations.

The following commands should be on a single line. For documentation formatting reasons, they may appear here on several lines.

Note If the company code contains special characters (!, #, @, \$, etc.), you must wrap the company code in double quotation marks. For example: `COMPANY_CODE="XXXXDKIHWKH@ORFXXXX"`.

Base Install using a Company Registration Code

```
msiexec /q /i C:\Users\UserFolderName\Desktop\installer_vista_win7_win8-32-3.3.0.953.msi /L*
log.txt COMPANY_CODE=XYZ
```

In this basic install example, no policy is specified; therefore, the sensors are assigned to either the Standard policy, or to a policy that a sensor group specifies (if sensor groups are defined and the sensors match the sensor group criteria).

Base Install into a Specific Policy

```
msiexec /q /i C:\Users\UserFolderName\Desktop\installer_vista_win7_win8-32-3.3.0.953.msi /L*
log.txt COMPANY_CODE=XYZ GROUP_NAME=Phase1
```

Using the GROUP_NAME (policy assignment option) assigns the sensor to the specified policy. To use sensor groups to determine a policy assignment, omit this option.

Note For Windows sensors 3.8+, we recommend that you replace GROUP_NAME with POLICY_NAME. GROUP_NAME will work, but POLICY_NAME is preferred for clarity.

Configure RepCLI Authenticated User AD Group

```
msiexec /q /i C:\temp\installer_vista_win7_win8-32-2.0.4.9.msi /L* log.txt COMPANY_CODE=XYZ
CLI_USERS=S-1-2-34-567
```

Note For more information about RepCLI, see [Managing Sensors by using RepCLI](#) in the User Guide.

Windows Sensor Log Files and Installed Services

The following log files and installed services reside on endpoints that have a Windows sensor installed.

Windows log files

Use the /L* log.txt command line option to obtain an MSI log that shows the Windows installation process. A confer-temp.log file is also generated in C:\Users\Username\AppData\Local\Temp that shows the sensor registration attempts to the cloud. These two log files are required for troubleshooting installation and update issues.

The Windows 3.6 sensor stores some log files in Program Files and some log files in ProgramData. Previous versions of the sensor stored logs in in the \Program Files\Confer\Logs\ directory. Carbon Black will continue to move all log files to ProgramData to align with Microsoft guidelines. You must have administrative privileges to access the log files in ProgramData.

- \Program Files\Confer\Logs\

- \ProgramData\CarbonBlack\Logs\

Windows installed services

- Main sensor service: RepMgr64.exe, RepMgr32.exe, Scanhost.exe (if local scanning is enabled)
- Utility: RepUtils32.exe, RepWmiUtils32.exe
- UI: RepUx.exe

Installing Windows Sensors on Endpoints by using Group Policy

To install sensors by using Group Policy, make a batch file to pass the parameters to an edited .msi file.

By default, Group Policy installs software on startup; therefore, you must reboot the endpoint to install the sensor.

Create a Microsoft Installer Transform (.MST) File

To create an .mst file, perform the following procedure.

Procedure

- 1 Sign in to the Carbon Black Cloud console.
- 2 On the navigation bar, click **Inventory** and then click **Endpoints**.
- 3 Click **Sensor Options** and then click **Download sensor kits**. Download the .msi file for the Windows sensor .
- 4 Download and install the Orca installer; see [https://msdn.microsoft.com/en-%20us/library/windows/desktop/aa370557\(v=vs.85\).aspx](https://msdn.microsoft.com/en-%20us/library/windows/desktop/aa370557(v=vs.85).aspx).
- 5 Right-click the .msi file that you downloaded in Step 3 and click **Edit with Orca**.
- 6 Click **Transform > New Transform**.
- 7 Create additional **Property** table entries. Under **Tables > Property**, right-click in a blank space and then click **Add row**.
- 8 Click **Property** and enter "COMPANY_CODE". Click **Value** and enter the company registration code for your organization. (See [Obtain a Company Registration Code](#).)
- 9 If you are installing in a VDI Environment, see [Chapter 4 Installing Sensors on Endpoints in a VDI Environment](#) for additional parameters.

- Click **Transform > Generate Transform** and save the file as an `.mst` file. Use a file name that is easily recognizable.

Note Carbon Black recommends that you create a verbose `.msi` install log file to help troubleshoot Group Policy installation or update issues.

Automatically Create a Windows Installer .MSI Log

Use this procedure to create an `.msi` log.

Procedure

- Open the Group Policy editor and expand **Computer Configuration > Administrative Templates > Windows Components**.
- Select **Windows Installer** and double-click **Logging** or **Specify the types of events Windows Installer records in its transaction log**, depending on the Windows version.
- Select **Enabled**.
- In the **Logging** textbox, type `voicewarmupx`.
- Click **Save Changes**.

This setting creates an `.msi` install log for all users in the GPO in the `c:\Windows\Temp\` folder on the system volume.

- To enable Windows Installer `.msi` log using the registry:
 - Open Regedit.
 - Go to registry key
`HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Windows\Installer`.
 - Set the **Logging** registry value to `voicewarmupx`.

Note If Group Policy is configured to automatically create a Windows Installer `.msi` log, the registry value `voicewarmupx` should match the value that is configured in Group Policy.

Install Sensors by using Group Policy

You can install sensors via Group Policy by using the `.msi` file that you previously created.

Procedure

- Click **Start > Administrative Tools > Group Policy Management**.
- Click **Software Settings > Software Installation > New > Package**.
- Select the `.msi` file that you downloaded in the [Create a Microsoft Installer Transform \(.MST\) File](#) procedure.
- Under **Deployment Method**, click **Advanced**.

- 5 Add a package name that identifies the sensor (for example, WinSensor34). For 32 bit `.msi` files only: in the **Deployment** tab, click **Advanced** and uncheck **Make this 32-bit x86 application available to Win64 machines**. Click **OK**.
- 6 Click the **Modifications** tab and click **Add**. Select the `.mst` file and click **Save**.
- 7 If you use a script to force a reboot to install software, run the script.
- 8 Check the Carbon Black Cloud console periodically to verify that sensor information is populating and that the sensors are checking in regularly.

Note

- The path to the `.msi` and `.mst` files must be available through a network share that is accessible from everywhere in your network, and to which everyone has at least read permissions.
 - For additional optional installation properties, see [Windows Sensor Supported Commands](#).
 - Active Directory does not support command line parameters. You must make a batch file to pass the parameters or package to an edited `.msi` file. Upon the next system restart, a drive is mounted and the installation is scheduled. The installation failure rate when using Active Directory is usually higher than with other software management tools.
 - By default, Group Policy installs software on startup. You can force an install/reboot by using a script. Consider the restart requirement when you deploy sensors via Group Policy.
-

Installing Windows Sensors on Endpoints by using SCCM

You can install Windows sensors by using System Center Configuration Manager (SCCM).

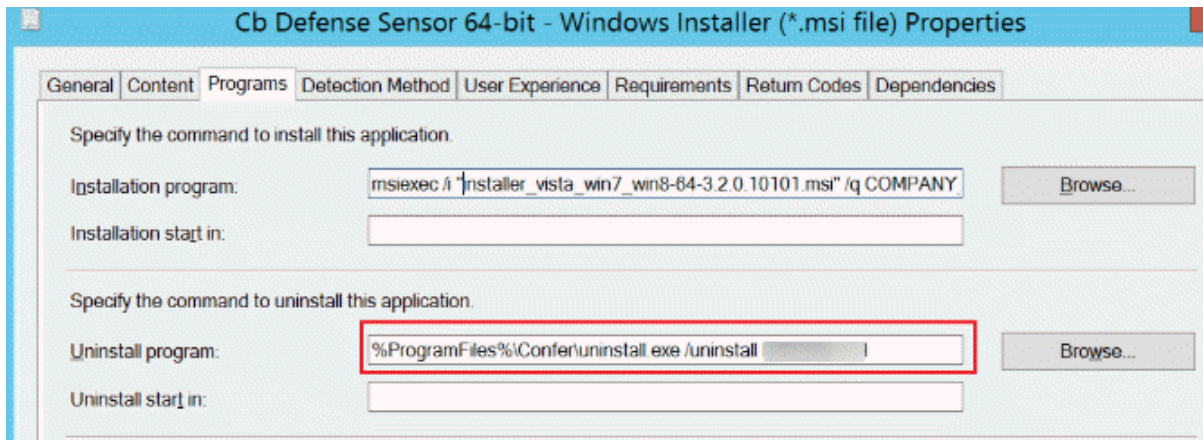
Add the Sensor Application to SCCM

As a first step in installing a Windows sensor by using SCCM, perform the following procedure.

Procedure

- 1 Open SCCM Configuration Manager.
- 2 In the **Software Library**, click **Overview > Application Management > Applications**.
- 3 Right-click **Applications** and click **Create Application**.
- 4 On the **General** page, select **Automatically detect information about this application from installation files**:
 - **Type**: Windows Installer (*.msi file)
 - **Location**: Accessible share that contains the sensor `.msi` file
- 5 Click **Next**. On the **Import Information** page, a message displays: **Application information successfully imported from the Windows Installer**. Click **Next**.

- 6 On the **General Information** page, add the required `COMPANY_CODE` install parameter and any other optional install parameters. See [Windows Sensor Supported Commands](#) for options. Click **Next** and on the **Summary** page, click **Next**.
- 7 On the **Completion** page, view the application details and click **Close**.
- 8 In the **Software Library**, right-click **Cb Defense Sensor Application** and click **Properties**.
- 9 Click the **Deployment Type** tab. Click the deployment type for CB Defense and click **Edit**. Note that the CB Defense type applies to Endpoint Standard, Enterprise EDR, and Audit and Remediation.
- 10 Click the **Programs** tab. If the **Require code to uninstall sensor** is enabled for the sensor policy and you want to be able to uninstall the sensor using SCCM, change the uninstall command from `msiexec /x "installer_vista_win7_win8-xx-x.x.x.xxxx.msi"` to `%ProgramFiles%\Confer\uninstall.exe /uninstall <Company Deregistration Code>`.



- 11 Click the **Detection Method** tab. Select the configured detection rule and click **Edit Clause**. Change the **Setting Type** to **File System**.
- 12 Select **The file system setting must satisfy the following rule to indicate the presence of this application**.
- 13 Set **Path** to `%ProgramFiles%\Confer` and **File or Folder name** to `RepUx.exe`.

- 14 Configure **MSI Property Version, Operator Greater than or equal to**. The **Version** is the currently installed sensor version.

Create a rule that indicates the presence of this application.

Setting Type: File System

Specify the file or folder to detect this application.

Type: File

Path: %ProgramFiles%\Confer Browse...

File or folder name: RepUx.exe

☐ This file or folder is associated with a 32-bit application on 64-bit systems.

☐ The file system setting must exist on the target system to indicate presence of this application

☒ The file system setting must satisfy the following rule to indicate the presence of this application

Property: Version

Operator: Greater than or equal to

Value: 3.2.0.10101

OK Cancel

- 15 Click **OK** three times to save **Detection Rule**, **Detection Method**, and **Deployment Type**.

Deploy the Sensor Application using SCCM

To install a Windows sensor by using SCCM, follow this procedure.

Procedure

- 1 Open SCCM Configuration Manager.
- 2 In the **Software Library**, click **Overview > Application Management > Applications**.
- 3 Select the CB Defense application, and click **Deploy**.
- 4 On the **General** page, for the **Collection** field, click **Browse**. From the dropdown menu, select **Device Collections** and select a collection of devices. Click **Next**.
- 5 On the **Content** page, click **Add** to add a distribution point. Click **Next**.

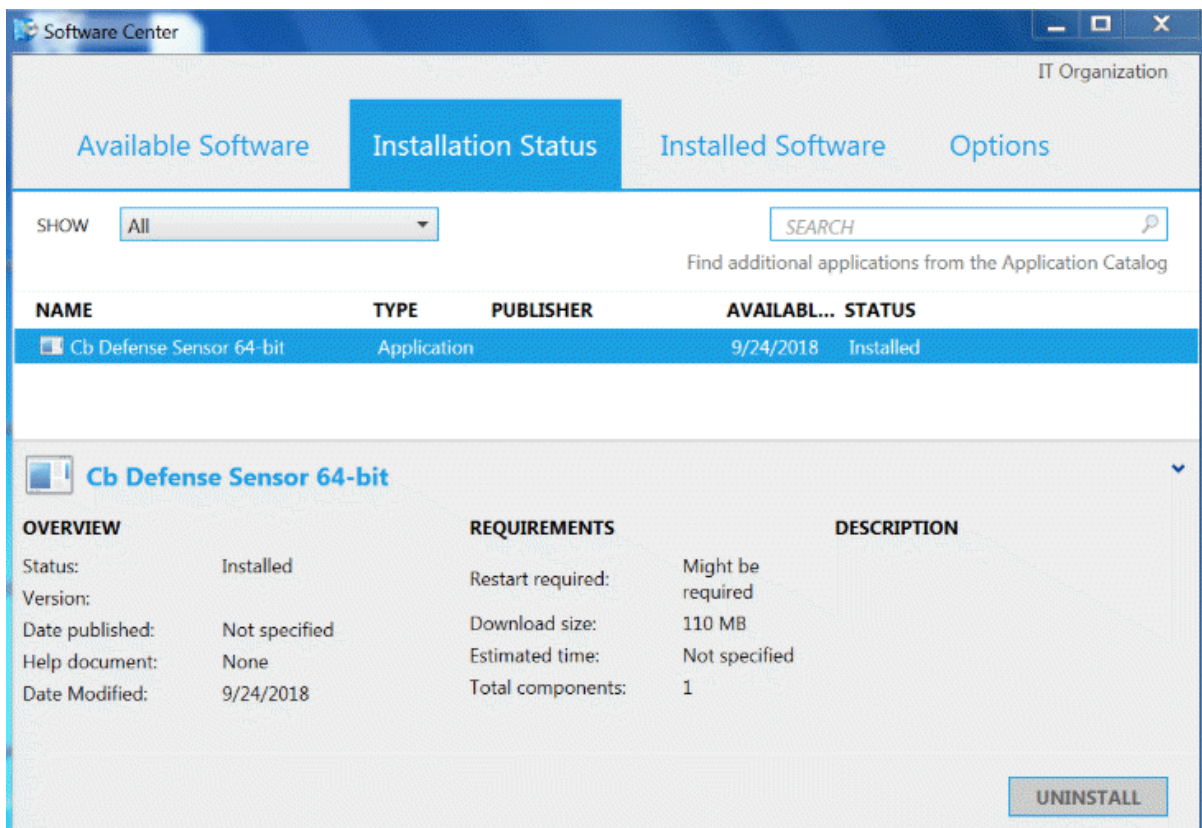
- 6 On the **Deployment Settings** page, set **Action** to **Install**, set **Purpose** to **Required**, and click **Next**.
- 7 On the **User Experience** page, set your deployment preferences and click **Next**.
- 8 On the **Alerts** page, set your alert preferences and click **Next**.
- 9 On the **Summary** page, review and confirm all settings and click **Next**.
- 10 On the **Completion** page, click **Close**.

Verify that the Sensor Application was Deployed via SCCM

After installing the Windows sensor by using SCCM, perform the following procedure to verify the deployment.

Procedure

- 1 In SCCM Configuration Manager, select the **CB Defense Sensor Application**, click the **Deployments** tab, and check **Compliance** status.
- 2 On the target device, open the Software Center and view the **Installation Status** or **Installed Software** tab.



Installing Carbon Black Cloud Sensor for Windows by Using Workspace ONE UEM

As a Workspace ONE administrator, you can deploy the Carbon Black Cloud sensor for Windows as a managed application with Workspace ONE UEM and thus, silently deploy the sensor across all your managed devices that are running Windows 10.

Procedure

- 1 [Deploy the Carbon Black Cloud Sensor for Windows as Managed Application in Workspace ONE UEM](#)

Perform the following procedure to add the Carbon Black Cloud sensor as an internal application and configure the deployment options in Workspace ONE UEM.

- 2 [Verify that Carbon Black Cloud Sensor for Windows Installed as Managed Application with Workspace ONE UEM](#)

You can use the Workspace ONE UEM admin console to verify that the Carbon Black Cloud sensor has been installed as a managed application on the assigned devices.

Deploy the Carbon Black Cloud Sensor for Windows as Managed Application in Workspace ONE UEM

Perform the following procedure to add the Carbon Black Cloud sensor as an internal application and configure the deployment options in Workspace ONE UEM.

Prerequisites

- Workspace ONE UEM with permissions to manage devices and applications.
- Carbon Black Cloud console access and admin account credentials.
- A device running Windows 10 to test the integration.
- Follow the instructions in [Obtain a Company Registration Code](#) and [Download Sensor Kits](#) to obtain your company registration code and download the Windows sensor kit.

Procedure

- 1 Open the Workspace ONE UEM admin console.
- 2 Add the Carbon Black Cloud sensor for Windows as an internal application by uploading the Carbon Black Cloud sensor MSI installation file.

- 3 Navigate to the **Deployment Options** tab and define how the Workspace ONE UEM must install the sensor application on the device.

The following options under the **How to Install** section are set by the Workspace ONE administrator and the rest of the options are set automatically by Workspace ONE UEM.

- a Populate the **Install Command** text box with the following command: `msiexec /i "installer_vista_win7_win8-64-3.5.0.1523.msi" /qn COMPANY_CODE=<REPLACE WITH YOUR REGISTRATION CODE> .`
- b Optional. To obtain the installation log file for troubleshooting purposes, add `/L*vx <file name>` to the above command.

```
msiexec /i "installer_vista_win7_win8-64-3.5.0.1523.msi" /qn /L*vx <file name>
COMPANY_CODE=<REPLACE WITH YOUR REGISTRATION CODE>
```

- c Set the **Admin Privileges** to **YES** if not so already.

The Carbon Black Cloud sensor requires admin privileges for installation.

- 4 Click the **Save & Assign** button.
- 5 Assign the Carbon Black Cloud sensor application to the **Assignment Groups**, which represent the devices that must have the sensor installed.

What to do next

Confirm that the sensor application installation completed successfully.

Verify that Carbon Black Cloud Sensor for Windows Installed as Managed Application with Workspace ONE UEM

You can use the Workspace ONE UEM admin console to verify that the Carbon Black Cloud sensor has been installed as a managed application on the assigned devices.

Procedure

- 1 Go to **Devices > List View**
- 2 Select a device and click the **Apps** tab.
- 3 Locate the Carbon Black Cloud Sensor 64-bit in the list of applications.

Results

The Carbon Black Cloud sensor is installed as a managed application on the devices that you previously assigned.

Search for Sensors

6

On the **Inventory > Endpoints** page in the Carbon Black Cloud console, you can search for specific sensors by any criteria that exists in the list of sensors. For example, you can search for specific devices, users, or operating systems.

The following table provides examples of valid operating system search queries. They are not case-sensitive.

Note Operating system versions listed in the following table are examples only; other operating system versions are accepted as well.

Table 6-1. Sensor Search by OS

| Linux | macOS | Windows |
|--------------------|-------------------------------|----------------|
| CentOS 7.9-2009 | MAC | Windows |
| RHEL 7.8 | OS X | Windows Server |
| Amazon 2.0 | 10.14.6 | Windows 10 |
| Debian 9.13 | 10.15.7 | x64 |
| Ubuntu 19.10 | 10.14.* where * is a wildcard | x86 |
| OpenSUSE Leap 15.2 | | |
| SLES 12 SP2 | | |

Updating Sensors on Endpoints

7

It is important that you keep your sensor versions up-to-date. There are several ways to update sensors.

- Update sensors on selected endpoints through the Carbon Black Cloud console. See:
 - [About Updating Sensors on Endpoints through the Console](#)
 - [Update Sensors on Endpoints through the Console](#)
- Reinstall the sensors.
- Use third party tools such as SCCM or GPO; see:
 - [Update Sensors on Endpoints that were Deployed by using SCCM](#)
 - [Update Sensors on Endpoints by using Group Policy](#)
- Update a sensor by double-clicking the new installer package or by issuing a command on the command line. Standard command line options are applicable. Command line options from the first install persist across updates.
 - To update a Windows sensor through the command line, see [Update Windows Sensors on Endpoints through the Command Line](#).
 - To update a Linux sensor through the command line, see [Update Linux Sensors on Endpoints through the Command Line](#).
 - To update a macOS sensor through the command line, simply reinstall the sensor. See [Chapter 3 Installing macOS Sensors on Endpoints](#) and [macOS Sensor Command Line Install](#).

This chapter includes the following topics:

- [About Updating Sensors on Endpoints through the Console](#)
- [Update Windows Sensors on Endpoints through the Command Line](#)
- [Update Sensors on Endpoints by using Group Policy](#)
- [Update Sensors on Endpoints that were Deployed by using SCCM](#)
- [Update Linux Sensors on Endpoints through the Command Line](#)
- [View Progress of Sensor Updates](#)

■ Sensor Status and Details

About Updating Sensors on Endpoints through the Console

You can add up to 10,000 sensors to an upgrade request (job) in the Carbon Black Cloud console. After a Sensor Update Status (SUS) job is created, it can remain in a pending state while other jobs are being processed.

Sensor updates are prioritized by the date of the request, from oldest to newest. When there are less than 500 sensors eligible for update in all currently processing jobs, oldest jobs are promoted first. If 500 upgrade slots are taken by the same job, Carbon Black can also pull in 10 sensors from a smaller job.

Note The completion of large update requests can be delayed if subsequent, smaller requests follow. Of the concurrent sensors available to update at a time, sensors from smaller requests are given priority for updates over larger processing requests.

An organization can have multiple 10,000 sensor update jobs at the same time.

The number of concurrent updates is the lesser of 25% of the total organization size or 500. For example, an organization that has 100 total sensors would hint up to 25 sensors to update at a time, and an organization that has 100,000 sensors would hint up to 500 sensors to update at a time. When an individual sensor completes its update process successfully or returns an error, a new sensor can be added to the processing queue to be updated.

The system attempts to upgrade up to 500 sensors at a time, and only considers sensors that are in a processing state (not pending). A job can only be promoted from pending to processing if at least one of its sensors has checked in within the last 30 minutes.

The processor runs every five minutes to see how many openings there are currently in the queue. It adds eligible sensors to the queue and sends hints for eligible sensors that are already in the queue. Sensors must have checked in within the last 30 minutes to be considered, and then must check in again after they are assigned a position in the queue.

SUS waits four hours before clearing any openings in a cancelled job. If a cancelled job had sensors, sensors that are in the processing state fill those openings in the queue.

Note Processing updates automatically timeout after two weeks. Timeouts occur even if the sensor has been hinted for an update, but the sensor has not successfully completed the update.

To monitor the status of sensor updates, see [View Progress of Sensor Updates](#).

Update Sensors on Endpoints through the Console

You can update sensors through the Carbon Black Cloud console.

Important If you are updating to the Windows 3.6 sensor, see [Configure a Firewall](#).

Procedure

- 1 Sign into the Carbon Black Cloud console.
- 2 On the navigation bar, click **Inventory** and then click **Endpoints**.
- 3 Search for and select the sensors to update.
- 4 Click **Take Action** and then click **Update Sensors**.
- 5 Confirm the number of sensors to update.

Tip To update more than 200 sensors, enter a search string, click **Actions > Update Sensors**, and select **Update sensor on all *X* Assets** where *X* is the number of devices that match the search. If the search returns more than 10,000 assets, this option does not display.

Update Sensors
✕

Are you sure you want to:

☐ Update sensors on the 1 selected assets
☒ Update sensors on all 224 assets

| PLATFORM | VERSION |
|-------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| Windows | <div style="border: 1px solid #ccc; padding: 2px 5px; display: flex; align-items: center;"> -- <div style="margin-left: 5px;">▼</div> </div> |
| macOS (10.14 - 10.15, 11, 12) | <div style="border: 1px solid #ccc; padding: 2px 5px; display: flex; align-items: center;"> -- <div style="margin-left: 5px;">▼</div> </div> |

Deploying a large number of sensors at once may strain your network. Deploy small batches of 10-20 sensors every hour, and repeat this until your deployment is complete.

Update

Cancel

- 6 Select the sensor version from the **Version** dropdown menu.
- 7 Select the checkbox to acknowledge that endpoints might be rebooted and then click **Update**

Note In limited cases, updates can cause endpoints to reboot.

Results

After you have initiated the sensor updates, you can view the progress of the updates on the **Sensor Update Status** tab on the Endpoints page. See:

- [Sensor Status and Details](#)
- [Sensor Filters](#)

■ View Progress of Sensor Updates

When a sensor update status displays **Completed**, a hyperlinked count becomes available in the **Updated** column. Click the hyperlinked count to open a new browser tab to the Endpoints page, where the sensors that successfully updated are shown. If any sensors did not update, a hyperlinked count displays in the **Errors** column. Click this link to open a new browser tab to the Endpoints page, and the sensors that did not update are shown.

If the **Updated** or **Errors** sensor count is greater than 500, the hyperlinks are disabled, and only the **Export** option is available under the **Actions** column. Click **Export** to generate and download a CSV file that contains the count details.

If any sensors failed to update, the **Sensor Update Status** tab displays the reason for the failure.

Update Windows Sensors on Endpoints through the Command Line

You can update Carbon Black Cloud Windows sensors through the command line.

Note Not all command line options apply to a Windows sensor update; see [Windows Sensor Supported Commands](#) for details.

Procedure

- 1 Sign into the endpoint.
- 2 Download the updated sensor file; see [Download Sensor Kits](#).
- 3 Open an administrative command prompt.
- 4 Run the update command; for example:

```
%SYSTEMROOT%\system32\msiexec.exe" /qN /i
"C:\temp\installer_vista_win7_win8-32-2.0.4.9.msi" /L*v+ "%SYSTEMROOT%\temp\cb-installer-
version.log
```

Note In this example, the `L*v+` option creates a verbose log in `c:\windows\temp\` that will append to any existing log rather than overwriting and replacing it.

Update Sensors on Endpoints by using Group Policy

If you deploy sensors by using Group Policy, you must remove the existing sensor from the current Group Policy before you can perform a sensor update using Group Policy, the Carbon Black Cloud console, SCCM, manual updates, etc.

Procedure

- ◆ Use one of the following procedures to remove a sensor from a Group Policy or update sensors using Group Policy?
 - To remove the existing sensor from Group Policy
 - a Click **Start > Administrative Tools > Group Policy Management** and select the Group Policy Object (GPO).
 - b Click **Computer Configuration > Policies > Software Settings > Software Installation**.
 - c Right-click the CB Defense Sensor package and click **All Tasks > Remove...**
 - d Select **Allow users to continue to use the software but prevent new installations** and click **OK**.

Note The previous procedure removes the `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Group Policy\AppMgmt\{Cb Defense GUID}` registry key without uninstalling the current version of the sensor. To confirm that the registry key is removed, open Regedit and go to `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Group Policy\AppMgmt`. Search for "CB Defense", "PSC Sensor", or "Carbon Black Cloud". If no results are found, the key is removed.

Note If you are updating from Windows sensor version 3.2.x.x, read [GPO upgrade fails on sensor version 3.2.x.x](#).

- To update sensors by using Group Policy
 - a Follow the preceding procedure to remove the sensor from its existing Group Policy.
 - b Force a Group Policy update on all endpoints.
 - c Use the following instructions to update the sensors: [Install Sensors by using Group Policy](#).

Update Sensors on Endpoints that were Deployed by using SCCM

If you deployed sensors by using System Center Configuration Manager (SCCM), you can configure SCCM to allow alternate methods of updating the sensors.

Procedure

- 1 Open SCCM and go to the **Software Library**.
- 2 Click **Overview > Application Management > Applications > Carbon Black**.
- 3 Click the **Deployment Type** tab and select the **Deployment Type** that is configured for the sensor.

- 4 Click the **Detection Method** tab, click the configured detection rule, and click **Edit Clause**.
- 5 Change the **Setting Type** to **File System**.
- 6 Set **Path** to `%ProgramFiles%\Confer`.
- 7 Set **File or Folder name** to `RepUx.exe`.
- 8 Select **The file system setting must satisfy the following rule to indicate the presence of this application**.
- 9 Configure **MSI Property Version** operator to **Greater than or equal to**. **Version** should be the currently installed sensor version.
- 10 Click **OK** three times to save the configuration.

Update Linux Sensors on Endpoints through the Command Line

You can update Carbon Black Cloud Linux sensors through the command line.

Procedure

- 1 Sign into the endpoint.
- 2 Download the updated sensor file; see [Download Sensor Kits](#).
- 3 Unpack the agent tar ball into: `*/var/opt/carbonblack/psc/pkgsg/upgrade_staging/*`. If you have not previously updated the sensor, this folder does not exist and you must create it.
- 4 Run the update script from `/var/opt/carbonblack/psc/pkgsg/upgrade_staging` location:

RPM:

```
$rpm -U cb-psc-sensor-xxx.rpm
```

Note For the RHEL sensors kit, you must specify the rpm package that corresponds to the distro version that you are installing.

el6 --> centos/rhel/oracle 6.0-6.x

el7 --> centos/rhel/oracle 7.0-7.x

el8 --> centos/rhel/oracle 8.0-8.x

DEB:

```
$dpkg --force-conffold -i cb-psc-sensor-xxx.deb
```

- 5 Verify the following:
 - Agent is upgraded - `/opt/carbonblack/psc/bin/cbagentd -v` to make sure that the agent matches the version you installed.

- Kernel or BPF module is loaded
 - Kernel module: Run the following command and verify that there is a 1 in the right column of the output. This shows that the kernel module is loaded and enabled. Other versions of the kernel might display as disabled; this is acceptable.

Command: `lsmod | grep event_collector`

Sample output: `event_collector_2_x_yyyyyy zzzzz 1`

- BPF module: Run the following command and verify that the grep returns a single result with the command `event_collector`. Command: `ps -e | grep event_collector`
- Check agent-blade details on the Endpoints page in the server console:
 - Updated agent details are displayed
 - Agent checks in with the server at regular intervals

View Progress of Sensor Updates

You can monitor the status of sensor updates on the **Sensor Update Status** tab on the Endpoints page.

To stop a processing or pending update request, click the **Stop** icon in the **Actions** column.

Sensor Update Status

The progress of a sensor update is indicated by the **Status** column, along with an accompanying progress bar.

- **Pending:** Update has been requested but has not begun to process; corresponds with the **Requested** column timestamp.
- **Processing:** Update is currently in progress; updates will automatically timeout after two weeks.
- **Completed:** All sensors in the update have either succeeded or failed; corresponds with the **Completed** column timestamp.
- **Stopped:** Update has been cancelled; stopped updates cannot be restarted. A new update must be initiated.

Note Processing updates automatically timeout after two weeks. Timeouts occur even if the sensor has been hinted for an update, but the sensor has not successfully completed the update. Typically, sensors that have not updated due to a timeout will show a "Sensor unresponsive" error. This indicates that the sensor could not be reached for an update within the two-week period.

View results of sensor updates

After an update begins to process, the number of successful or failed sensor updates begin to populate in the table in the **Updated** and **Errors** columns. When completed, the sum of successful updates and any failed updates match the initial number of sensors requested for update in the **Sensors** column.

View Updated Sensors

Click the hyperlinked number of successfully updated sensors in the **Updated** column to view the updated sensors on the **Endpoints** tab. A hyperlink only appears if an update request is either **Completed** or **Stopped** and if the number of updated sensors is fewer than 500.

Export Results

In the **Actions** column, click the **Export** icon to download a CSV file of any **Completed** or **Stopped** update request.

Use the CSV file to view full results of updates. The file contains useful information about your updates, including the Device IDs of all requested sensors, their initial and updated sensor versions, and the reason for any update failure.

View Failed Sensors and Errors

Click the hyperlinked number of failed sensors in the **Errors** column to view the failed sensors on the **Endpoints** tab. A hyperlink only appears if an update request is either **Completed** or **Stopped** and if the number of failed sensors is fewer than 500.

If an update contains failures, click the caret on the left of the row in the table to view a summary of failure reasons. Sensors can fail due to:

- **Sensor unresponsive:** The sensor was offline or failed to check in with the system during the timeframe of the update.
- **No sensor found:** The sensor could not be found. The sensor is probably deregistered.
- **Update stopped by user:** The update request was stopped by a user before the sensor could update.
- **Update error:** The sensor failed to update to the targeted version.

| Column | Description |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Requested | Date and time of the initial update request. |
| Completed | Date and time of the finished update. An update can show this status even if it contains both successful and failed sensor updates. |
| Status | Progress of a sensor update. The status of an update can be: Pending, Processing, Completed, or Stopped. |
| Sensors | Total number of sensors requested for an update. |
| Updated | Number of successfully updated sensors. This number will change as more sensors are successfully updated until the update has completed or has been stopped. |

| Column | Description |
|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Errors | The number of sensors that have failed to update. This number will change as more sensors fail to update, until the request has completed or has been stopped. |
| Actions | Click the Stop icon to stop a processing or pending request. When updates are completed or stopped, click the Export icon to download a CSV file to view the full results of the update request. |

Sensor Status and Details

The Endpoints page in the console displays sensor status and details.

The **Endpoints** tab on the Endpoints page, displays all deployed sensors by default.

ENDPOINTS

Initial sensors on endpoints and use sensor groups to automatically assign policies to sensors

Endpoints

Sensor Update Status

Sensor Options

Add Group

FILTERS

Clear

Search

Export

+ Status (9)

+ Sensor Version (2)

+ OS (2)

+ Signature Status (4)

+ Policy (12)

+ Golden Image Status (2)

+ Sensor Group (1)

STATUS

NAME

USER

OS

GROUP/POLICY

S...

SENSOR

T

LAST CHECK-IN

ACTIONS

You can limit which sensors to display by using the **Filters** options in the left pane. See [Sensor Filters](#).

To export the table data into a CSV file, click the **Export** button in the upper right section of the page.

You can define which columns display in the results table. Click **Configure Table** at the bottom of the page to hide or display columns.

The resulting sensor data displays in the following columns by default:






Status

The **Status** column indicates the state of a sensor and any administrator actions that have been taken on the sensor. This column can contain multiple icons to indicate the sensor state.

Table 7-1. Sensor Status

| Icon | Status | Description |
|------|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | Active | Sensor has checked in within the last 30 days. |
| | Bypass | Sensor has been put into Bypass mode by an administrator. All policy enforcement on the device is disabled and the sensor does not send data to the cloud. Sensors also enter Bypass mode briefly during a sensor update. See Bypass Reasons . |

Table 7-1. Sensor Status (continued)

| Icon | Status | Description |
|-----------------------------------------------------------------------------------|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Deregistered | Sensor has been deregistered or uninstalled; it will persist on the Endpoints page in this state until it is removed. |
|  | Errors | Sensor is reporting errors. |
|  | Inactive | Sensor has not checked in within the last 30 days. |
|  | Pending install | Sensor has not been installed following an installation request email sent to a user. |
| No icon | Pending update | Sensor is pending an update. |
|  | Quarantine | <p>Sensor has been put into Quarantine mode. It is isolated from the network to mitigate the spread of potentially malicious activity.</p> <p>Note Quarantine is not supported for Linux sensors before version 2.13.</p> |
| No icon | Sensor out of date | Sensor is not using the current available sensor release version and is eligible for update. |

Name

The **Name** column represents the Device ID of the endpoint.

User

The **User** column displays user data based on the OS and the sensor version.

- macOS 3.3.2+ versions display the last active user logged in to the device.
- Windows 3.5+ versions display the last active user logged in every 8 hours; if there is no interactive user logged in within the 8 hour window, a noninteractive user name can appear.
- Previous macOS and Windows versions display the user who installed the sensor.
- Linux versions are intentionally left blank because multiple, simultaneous logged-in users and desktop users are possible.

OS

The **OS** column lists the operating system that is running on the endpoint.

Group/Policy




The **Group/Policy** column lists the group to which the sensor belongs (if any), how its policy was assigned, and the name of the assigned policy. If a sensor is not a member of a sensor group and was manually assigned a policy, it is listed as **Manually assigned**. If the sensor metadata does not match any group criteria, it is listed as **Unassigned**.

Signature

The **Signature** column displays an icon that represents the status of each sensor signature version.

Note This feature is only available for Windows sensors.

Table 7-2. Signature Version Status

| Icon | Status |
|------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Signature version is current. The installed signature version was released within 7 days of the current date. |
|  | Signature version is out of date. The installed signature version has not been released within 7 days of the current date. |
|  | Signature version is not yet reported or is unidentifiable. Signatures can display as not reported if the local scan is not configured or if the sensor encountered an error after the local scan was configured. |
| No icon | Unidentifiable sensor signature version. This presents for macOS and Linux sensors. |

Sensor

The **Sensor** column lists the sensor version that is running on the endpoint.

Target

The **Target** column lists the target value of the endpoint. This value can be Critical, High, Medium, or Low.

Last Check-in

The **Last Check-in** column displays the last time and date that the sensor checked in with the Cloud.

Actions

The **Actions** column provides two actions that you can perform on the endpoint.



Click the  icon to investigate any events that have occurred on the endpoint.

Click the > icon to open an **Endpoint Details** pane that provides more details about the selected endpoint.

Endpoint details

Device ID

Internal IP

External IP

Registered7:30:18 pm Nov 29, 2021

Last check-in7:30:19 pm Nov 29, 2021

Signature Version:

Installed by

Live response status

OSWINDOWS

OS version

Uninstall code

Actions

Update sensor

Update the sensor on on

Quarantine asset

Isolate on network to mitigate risk

Enable Bypass

Disable policy enforcement on

Update

Quarantine

Enable Bypass

Sensor Filters

You can define which sensors display on the **Endpoints** tab on the Endpoints page.

The following sensor filters are available:

Status

You can filter sensors by status. For more information about status conditions, see [Sensor Status and Details](#). Status filters are:

- Active
- Bypass
- Deregistered
- Errors
- Inactive
- Pending install

- Pending update
- Quarantine
- Sensor out of date

Sensor Version

You can select which sensor versions to display, or display all versions.

OS

You can filter sensors based on the device operating system, such as macOS or Windows.

Signature Status

For Windows sensors, the status of the local scan signature version displays in the **Sig** column on the Endpoints page. Possible filters are:

- Not Available: The sensor signature version is not yet reported.
- Not Applicable: Unidentifiable sensor signature version. This presents for macOS and Linux sensors.
- Out of date: The sensor signature files show as out-of-date seven days after being disabled until the updates are reenabled.
- Up to date: The sensor signature files are up-to-date if the installed signature version is released within seven days of the current date.

Policy

You can select the sensors that display based on their assigned policy.

Golden Image Status

You can filter the displayed sensors (endpoints) based on their type: as not a golden image, or as a golden image with clones.

Sensor Group

You can display sensors based on their assigned sensor group.

Bypass Reasons

You can view the reason an asset goes into a bypass mode in the Carbon Black Cloud console.

The following table lists the possible reasons for an asset to go in a bypass mode, and the remediation actions that you can perform. You can use a search value associated with a bypass reason to filter assets matching the bypass reason.

| Search value of the bypass reason | Display value of the bypass reason | Description | Action to resolve bypass |
|------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| sensorStates:"CSR_ACTION" | Bypass (Admin action) | The Carbon Black Cloud console instructs the sensor to go into a bypass mode. Relates to sensors supporting Windows, macOS, and Linux. | Use the Carbon Black Cloud console or a local action to remove the sensor from the bypass state. |
| sensorStates:"REPUX_ACTION" | Bypass (Local action) | A local action instructs the sensor to go into bypass mode. For example, enable bypass locally on the sensor: <ul style="list-style-type: none"> ■ By elevating a command prompt and executing the command <code>"C:\Program Files\Confer\Uninstall.exe" /bypass 1 <UninstallCode></code> ■ By logging into the asset with credentials configured at sensor installation, launching a command prompt, and executing the command <code>repcli bypass 1</code> from the directory <code>C:\Program Files\Confer</code>. ■ By using the policy setting "Allow user to disable protection". For details on this setting, see General Policy Settings in the user guide. ■ By executing the command for installing the sensor with the option <code>bypass=1</code> in its syntax. Relates to sensors supporting Windows, macOS, and Linux. | Use the Carbon Black Cloud console or a local action to remove the sensor from the bypass state. |
| <ul style="list-style-type: none"> ■ sensorStates:"UNSUPPORTED_OS" OR ■ sensorStates:"OS_VERSION_MISMATCH" | Bypass (Unsupported OS) | The installed sensor does not support the operating system. Relates to sensors supporting macOS and Linux. | Upgrade the sensor or the operating system to a supported version. For information on the product operating environment requirements, see VMware Carbon Black Cloud Documentation . |
| <ul style="list-style-type: none"> ■ sensorStates:"DRIVER_LOAD_NOT_GRANTED AND ■ sensorStates:"DRIVER_USERSPACE" | Bypass (System ext. not approved) | The Carbon Black Cloud macOS sensor's System Extension is not approved. Relates to sensors supporting macOS. | Approve the System Extension that the sensor utilizes. See Approving the System Extension and Network Extension for macOS 11+ in the sensor installation guide. |
| <ul style="list-style-type: none"> ■ sensorStates:"DRIVER_LOAD_NOT_GRANTED" AND ■ sensorStates:"DRIVER_KERNEL" | Bypass (Kernel ext. not approved) | The Carbon Black Cloud macOS sensor requires a Kernel Extension approval, regardless of the previous Kernel Extension approval status. Relates to sensors supporting macOS. | Approve the Kernel Extension. See Approve the Kernel Extension (macOS 10.13 – macOS 11) in the sensor installation guide. |

| Search value of the bypass reason | Display value of the bypass reason | Description | Action to resolve bypass |
|---------------------------------------------|------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| sensorStates:"REMGR_INIT_ERROR" | Bypass (Service Error) | The sensor is having a problem connecting to the event_collector. Relates to sensors supporting Linux. | Check that the Linux distribution is supported. For version compatibility, see VMware Carbon Black Cloud Linux Sensor Operating Environment Requirements . If the distribution is supported, contact VMware Carbon Black Support. |
| sensorStates:"KERNEL_HEADERS_NOT_INSTALLED" | Bypass (Contact support) | The Extended Berkeley Packet Filter (eBPF) implementation requires installation of the Linux kernel headers for the active kernel before sensor installation. Also, the sensor might be running an unsupported OS Kernel version. Relates to sensors supporting Linux. | Verify that the kernel headers are installed. See Prerequisites for Linux 4.4+ Kernels for Linux Sensor Versions 2.10+ in the sensor installation guide. For version compatibility, see VMware Carbon Black Cloud Linux Sensor Operating Environment Requirements . |
| sensorStates:"DRIVER_INIT_REBOOT_REQUIRED" | Bypass (Reboot required) | The asset requires a reboot to initialize the driver. Relates to sensors supporting macOS. | If a reboot does not resolve this, contact VMware Carbon Black Support. |
| sensorStates:"DRIVER_LOAD_PENDING" | Bypass (Extension load pending) | Loading extension is pending. Relates to sensors supporting macOS. | If a reboot does not resolve this, contact VMware Carbon Black Support. |
| sensorStates:"DRIVER_INIT_ERROR" | Bypass (Extension Error) | Driver fails in loading properly. Relates to sensors supporting Windows, macOS, and Linux. | If a reboot does not resolve this, contact VMware Carbon Black Support. |
| sensorStates:"SENSOR_UPGRADE_IN_PROGRESS" | Bypass (Update in progress) | The asset is going through a sensor update. Relates to sensors supporting Windows. | Resolves immediately after the sensor update completes. |
| N/A | Bypass (Contact Support) | Device is in bypass for an unknown reason. | Contact VMware Carbon Black Support for additional assistance. |

Uninstalling Sensors from Endpoints



You can uninstall sensors from the Carbon Black Cloud console or directly at the endpoint.

This chapter includes the following topics:

- [Uninstall Sensors from the Endpoint by using the Carbon Black Cloud Console](#)
- [Require Codes to uninstall Sensors at an Endpoint](#)
- [Uninstall a Linux Sensor from an Endpoint](#)
- [Uninstall a 3.5+ macOS Sensor from an Endpoint](#)
- [Uninstall a pre-3.5.1 macOS Sensor from an Endpoint](#)
- [Uninstall a Windows sensor from an Endpoint](#)
- [Delete Deregistered Sensors from Endpoints](#)

Uninstall Sensors from the Endpoint by using the Carbon Black Cloud Console

You can uninstall macOS and Windows sensors via the Carbon Black Cloud console.

Note You cannot uninstall Linux sensors via the Carbon Black Cloud console. You must uninstall Linux sensors by using the command line as explained in [Uninstall a Linux Sensor from an Endpoint](#).

Procedure

- 1 Sign in to the Carbon Black Cloud console.
- 2 On the navigation bar, click **Inventory** and then click **Endpoints**.
- 3 Search for and select the sensors to uninstall.
- 4 Click **Take Action** and then click **Uninstall**.

Results

After you uninstall a sensor, it persists on the **Endpoints** page as a deregistered sensor until you delete it. See [Delete Deregistered Sensors from Endpoints](#).

Require Codes to uninstall Sensors at an Endpoint

If you have deployed v3.1 or later sensors, you can protect the action of uninstalling the sensor at the endpoint by requiring a unique, randomly-generated code. This setting is enabled per policy, and is recommended for security purposes. The uninstall code is case-sensitive.

Procedure

1 To require a code to uninstall a sensor at an endpoint:

- a Sign in to the Carbon Black Cloud console, click **Enforce**, and then click **Policies**.
- b Select the policy.
- c On the **Sensor** tab, select the **Require code to uninstall sensor** checkbox and then click **Save**.

After you have enabled this setting, a user must have an individual device uninstall code or a company deregistration code to uninstall the sensor at the endpoint. No code is required to uninstall sensors from within the Carbon Black Cloud console.

An individual device uninstall code is automatically generated when a sensor is registered with the Carbon Black Cloud.

2 To view a sensor uninstall code at an endpoint:

- a Sign in to the Carbon Black Cloud console.
- b On the navigation bar, click **Inventory** and then click **Endpoints**.
- c Click the > next to the sensor to view the uninstall code.

You can also generate a company deregistration code, and use this code to uninstall any sensor in your organization.

Caution The company deregistration code can be used to uninstall all sensors in your organization. If you do not want this capability, do not generate the company deregistration code.

3 To generate a company deregistration code:

- a Sign in to the Carbon Black Cloud console.
- b On the navigation bar, click **Inventory** and then click **Endpoints**.
- c Click **Sensor Options** and then click **Company codes**.
- d Under **Company Deregistration Code**, click **Generate New Code**.

Note Only macOS and Windows sensors can be uninstalled with a company deregistration code. [Uninstall a Linux Sensor from an Endpoint](#) by using the command line.

Uninstall a Linux Sensor from an Endpoint

You can use this procedure to uninstall a Linux Sensor from an endpoint.

Note After you run the command, the sensor remains listed in the **Registered Devices** list on the **Endpoints** page in the console until you click **Take Action > Uninstall**.

Run the following command from the location where the installer kit was unpacked:

- For CentOS, RHEL, SUSE or Amazon Linux: `$ sudo rpm -e cb-psc-sensor`
- For Ubuntu: `$ sudo dpkg --purge cb-psc-sensor`

Note After you uninstall the Linux sensor on the following OS distributions: CentOS, RHEL, Suse, Amazon Linux, the sensor log files together with a copy of `cfg.ini` and directories such as `/opt/carbonblack` and `/var/opt/carbonblack` are not deleted from the endpoint.

Uninstall a 3.5+ macOS Sensor from an Endpoint

Use the following procedures to uninstall 3.5+ macOS sensors from endpoints.

Procedure

- ◆ Perform one of the following uninstall procedures:
 - To perform a command line uninstall of the sensor:
 - a Run the following command: `sudo /Applications/VMware\ Carbon\ Black\ Cloud/uninstall.bundle/Contents/MacOS/uninstall -y`
 - b If the **Require code to uninstall sensor** option is enabled, run the following command: `sudo /Applications/VMware\ Carbon\ Black\ Cloud/uninstall.bundle/Contents/MacOS/uninstall -y -c <Uninstall Code>`
 - c If the sensor was installed in KEXT mode, you must reboot the endpoint to fully remove the unloaded KEXT.
 - To perform an attended uninstall of the sensor:
 - a Mount the `CBCloud.dmg` and double-click **CBCloud Uninstall**.
 - b Proceed through the uninstallation prompts. You must authenticate as admin.
 - c If the sensor was installed in KEXT mode, you must reboot the endpoint to fully remove the unloaded KEXT.

Uninstall a pre-3.5.1 macOS Sensor from an Endpoint

Use this procedure to uninstall pre-3.5.1 sensors from a macOS endpoint.

By default, this mode is interactive and requires a confirmation prompt unless you specify the `-y` parameter. To view all command line parameters, run the command by specifying the `-h` parameter.

Procedure

- 1 Open **Terminal** with elevated privileges.
- 2 Type `sudo /Applications/Confer.app/uninstall -y` and click **Enter**.

If you require a device uninstallation code or a company deregistration code, enter it as part of the command; for example:

```
sudo /Applications/Confer.app/uninstall -y -c 35BQCCYX
```

Uninstall a Windows sensor from an Endpoint

This procedure describes how to uninstall a Windows sensor from an endpoint.

Note You can uninstall multiple sensors by using batch files or system management tools.

Procedure

- 1 Open a command prompt window with administrative privileges.
- 2 Go to the `Confer` directory.
- 3 Run the following command; if you require a device uninstallation code or a company deregistration code, enter it as part of the command; for example: `uninstall.exe /uninstall 35EQCCYG`

Results

The `Confer` directory and log files remain after the sensor is uninstalled. The `Confer` directory is removed after an uninstall and a reboot.

Uninstall Windows Sensors from an Endpoint by using Group Policy

You can use Group Policy to uninstall Windows sensors by following this procedure.

Procedure

- 1 Click **Start > Administrative Tools > Group Policy Management** and go to **Software Installation**.
- 2 In the **Results** pane, right-click the CB Defense Sensor application, click **All Tasks**, and then click **Remove**.
- 3 In the **Remove Software** dialog box, select **Immediately uninstall the software from users and computers** and click **OK**.

Results

The application is removed the next time a user logs on or restarts the computer.

Caution The sensor does not support uninstall using Group Policy if "Require code to uninstall sensor" is enabled. See <https://community.carbonblack.com/t5/Knowledge-Base/PSC-Sensor-uninstalled-without-de-registration-code/ta-p/84736>.

You can also uninstall a Windows sensor by using Group Policy **Software Installation > Results > Deployment**; however Carbon Black does not recommend this option.

Enable SCCM to Uninstall a Windows Sensor from an Endpoint

You can enable SCCM to uninstall a Windows sensor.

On the **Programs** tab in SCCM, if the **Require code to uninstall sensor** is enabled for the sensor policy and you want to uninstall the sensor using SCCM, change the uninstall command from `msiexec /x"installer_vista_win7_win8-xx-x.x.x.xxxx.msi"` to `%ProgramFiles%\Confer\uninstall.exe /uninstall <Company Deregistration Code>`.

Delete Deregistered Sensors from Endpoints

Use this procedure to delete deregistered sensors.

Procedure

- ◆ Use one of the following procedures to delete deregistered sensors from an endpoint:
 - **To manually delete deregistered sensors on an endpoint**
 - a Sign in to the Carbon Black Cloud console.
 - b On the navigation bar, click **Inventory** and then click **Endpoints**.
 - c Filter the list of sensors to show only devices that have deregistered sensors.
 - d Select the sensors to delete.
 - e Click **Take Action** and then click **Delete deregistered devices**. You are prompted to confirm the deletion.
 - **To automatically delete deregistered sensors on an endpoint**
 - a Sign in to the Carbon Black Cloud console.
 - b On the navigation bar, click **Inventory** and then click **Endpoints**.
 - c Click **Sensor Options** and then click **Sensor settings**.
 - d Select **Delete sensors that have been deregistered for** and set the time frame. Click **Save**.

Managing Sensors for VM Workloads

9

You can secure VMware workloads in your data center using Carbon Black Cloud. VMware workloads require Windows 3.6+ and Linux 2.9+ sensor versions.

This chapter includes the following topics:

- [Installing Sensors on VM Workloads](#)
- [Update Sensors for Workloads from the Console](#)
- [Update Linux Sensors on Workloads through the Command Line](#)
- [Uninstall Linux Sensors from Workloads](#)
- [Uninstall Windows Sensors from Workloads](#)
- [Delete Deregistered Sensors from Workloads](#)

Installing Sensors on VM Workloads

You install sensors on eligible VM workloads from the Carbon Black Cloud console.

The Carbon Black Cloud console allows you to view which deployed Virtual machine (VM) workloads in your data center are available for sensor installation. This data is available in the **Eligibility** column, part of the **Inventory > VM Workloads > Not Enabled** tab. The **Eligibility** column contains also the workloads that are not eligible for sensor installation and the once that need to upgrade to a supported OS version.

| Eligibility Column | Description |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Eligible | The eligible VM workloads have the appropriate version of the VMware Tools with the Carbon Black launcher and you can install sensors on them. |
| Not eligible | <p>VM workloads not eligible for sensor installation have the required version of the VMware Tools or Carbon Black launcher unavailable. To minimize your deployment efforts, a lightweight Carbon Black launcher is available with the VMware Tools. Carbon Black launcher must be available on the Windows and Linux virtual machines (VMs).</p> <ul style="list-style-type: none"> ■ For Windows VMs, the Carbon Black launcher is packaged with the VMware Tools. To receive the launcher for your workloads, you must install or upgrade VMware Tools to version 11.2 or later. ■ For Linux VMs, you must manually install the launcher available at VMware Tools Operating System Specific Packages (OSPs). To learn more, visit Carbon Black Cloud Workload Guide. <p>After the launcher is available, you can proceed to install sensors on your workloads inventory.</p> <ul style="list-style-type: none"> ■ If VMs are offline, you cannot proceed with the installation. Go to the vCenter Server and power on the VMs. |
| Not supported | Carbon Black Cloud Workload Plug-in does not support the Operating System (OS) or the OS version. Upgrade to the supported OS or version as per the system requirements. |

Prepare Your Workloads Environment for Sensor Installation

To prepare your environment for installing sensors on your deployed VM workloads you register the Carbon Black Cloud Workload Appliance with the vCenter Server and connect the appliance to the Carbon Black Cloud. Carbon Black launcher must be available on the VMs.

Procedure

- 1 Set up your [Carbon Black Cloud Workload appliance](#).

Carbon Black Cloud Workload Appliance must be online and connected to the Carbon Black Cloud via an API key to receive a sensor. You confirm appliance connectivity in two ways:

- In the Carbon Black Cloud console, check for available VM workloads on the **Inventory > VM Workloads > Not Enabled** tab.
- In the Carbon Black Cloud console, go to the **Settings > API Access > API Keys** page and click the appliance name to view connection status.

- 2 Enable Carbon Black Cloud through a lightweight Carbon Black launcher to install a sensor for VM workloads.

- For Windows VMs, Carbon Black launcher is packaged with [VMware Tools](#). You must install or upgrade VMware Tools to version 11.2.0 or later to obtain the launcher.

- For Linux VMs, you must manually install the launcher from VMware Tools Operating System Specific Packages (OSPs). Download and install Carbon Black launcher for your guest operating system from the package repository at <http://packages.vmware.com/>. For detailed instructions, see [Carbon Black Launcher for Linux VMs](#).

Install Sensors on VM Workloads

Use this procedure to install sensors on VM workloads through the Carbon Black Cloud console. You can use the configuration file to specify the proxy server that a Carbon Black launcher and a Carbon Black sensor can use after the installation completes.

Prerequisites

- Make sure you have configured firewall correctly. For firewall information, see [Configure a Firewall](#).
- For details on the command line installation options, see [Windows Sensor Supported Commands](#).
- The only supported proxy connection for the Carbon Black launcher and the Carbon Black sensor is the unauthenticated HTTP tunneling proxy.
- To obtain the Carbon Black launcher for Windows VMs with proxy support, install or upgrade VMware Tools to version 11.3.0 or later.

Procedure

- 1 Sign in to the Carbon Black Cloud console.
- 2 On the navigation bar, select **Inventory > VM Workloads**.
- 3 Click the **Not Enabled** tab and select eligible workloads.

Eligible workloads are running a supported OS and have a correct version of the VMware Tools with the Carbon Black launcher.

| Enabled Not Enabled Sensor Update Status | | | | | | |
|-----------------------------------------------------|-----------------------------------|----------------|-----------------------|----------------------------------------|--------------|-------------------------|
| FILTERS | Search | | | | | |
| | ELIGIBILITY | INSTALL STATUS | NAME | OS | VMWARE TOOLS | ADDED |
| | <input type="checkbox"/> Eligible | Not started | Server-101-vmware-001 | Microsoft Windows 10 (64-bit) | 11333 | 6:23:35 am Aug 9, 2021 |
| | <input type="checkbox"/> Eligible | Not started | Server-102-vmware-002 | Microsoft Windows Server 2012 (64-bit) | 11328 | 6:19:36 am Mar 16, 2021 |

- 4 Click the **Take Action** drop-down menu and select **Install sensors**.

5 Select the sensor version to install.

Install Sensors [X]

Install sensors on 1 selected workload(s)

SENSOR VERSION
Learn more in [Sensor Release Notes](#) and [Sensor Install Guide](#)

| OS | SENSOR VERSION |
|----------------|----------------|
| Windows 64-bit | 3.7.0.1375 ▼ |

SENSOR CONFIGURATION FILE
[Download a template](#) | Learn more in [Sensor Install Guide](#)
Authenticated proxy is not supported as part of configuration setting

[Upload File](#) (file format: ini, txt, conf, cfg)

Install **Cancel**

6 Optional. Update the sensor configuration file with proxy settings.

The configuration file tells both the Carbon Black sensor and the Carbon Black launcher what proxy to use.

- Click the **Download a template** link to use a sample configuration file. The company registration code and the Carbon Black Cloud URL are pre-populated in the template.
- Add the proxy server by specifying the server name and port number in the configuration file.
HTTPS is not supported.
- Click **Upload File** to upload the sensor configuration file that contains command line installation options such as the proxy configuration information.

7 Click **Install**.

You see a **Sensor installation submitted** notification and the install status for the VM changes to **In Progress**.

It takes up to 5 minutes for the installation to complete.

Results

After the sensor installs, it appears on the **Enabled** tab.

Update Sensors for Workloads from the Console

Use this procedure to update sensors for Workloads from the Carbon Black Cloud Console.

It is important that you keep your sensor versions up-to-date.

There are two ways to update sensors:

- You can update sensors on selected workloads through the console. You can select up to 10,000 sensors to update at one time. After you initiate sensor updates, the selected sensors receive the message to update the next time that they check in with the Carbon Black Cloud backend. The system allows up to 500 concurrent updates. When an individual sensor completes its update process, a new sensor is signaled to start its update.
- You can reinstall the sensors.

Procedure

- 1 Sign into the Carbon Black Cloud console.
- 2 On the navigation bar, click **Inventory** and then click **Workloads**.
- 3 Search for and select the sensors to update.
- 4 Click **Take Action** and then click **Update Sensors**.
- 5 Confirm the number of sensors to update.
- 6 Select the sensor version from the **Version** dropdown menu.
- 7 Click **Update**

Update Linux Sensors on Workloads through the Command Line

You can update Linux sensors through the command line.

Procedure

- 1 Sign into the workload.
- 2 Unpack the agent tar ball into: `*/var/opt/carbonblack/psc/pkgs/upgrade_staging/*`. If you have not previously updated the sensor, this folder does not exist and you must create it.
- 3 Run the update script from the `/var/opt/carbonblack/psc/pkgs/upgrade_staging` location:

RPM:

```
$rpm -U cb-psc-sensor-xxx.rpm
```

Note For the RHEL sensors kit, you must specify the rpm package that corresponds to the distro version that you are installing.

el6 --> centos/rhel/oracle 6.0-6.x

el7 --> centos/rhel/oracle 7.0-7.x

el8 --> centos/rhel/oracle 8.0-8.x

DEB:

```
$dpkg --force-confold -i cb-psc-sensor-xxx.deb
```

4 Verify the following:

- Agent is updated - `/opt/carbonblack/psc/bin/cbagentd -v` to make sure that the agent matches the version you installed.
- Kernel or BPF module is loaded
 - Kernel module: Run the following command and verify that there is a 1 in the right column of the output. This shows that the kernel module is loaded and enabled. Other versions of the kernel might display as disabled; this is acceptable.

Command: `lsmod | grep event_collector`

Sample output: `event_collector_2_x_yyyyyy zzzzz 1`

- BPF module: Run the following command and verify that the grep returns a single result with the command `event_collector`.

Command: `ps -e | grep event_collector`

Sample output: `85150 ? 00:00:05 event_collector`

- Check agent-blade details on the Workloads page in the server console:
 - Updated agent details are displayed
 - Agent checks in with the server at regular intervals

Uninstall Linux Sensors from Workloads

Use this procedure to use the command line to uninstall Linux sensors from Workloads.

Note After you run the following command, the sensor remains listed in the **Registered Devices** list on the **Workloads** page in the console until you click **Take Action > Uninstall**.

Run the following command from the location where the installer kit was unpacked:

- For CentOS, RHEL, SUSE or Amazon Linux: `$ sudo rpm -e cb-psc-sensor`
 - For Ubuntu: `$ sudo dpkg --purge cb-psc-sensor`
-

Results

After you uninstall a sensor, it persists on the **Workloads** page as a deregistered sensor until you delete it. See [Delete Deregistered Sensors from Workloads](#)

Uninstall Windows Sensors from Workloads

You can uninstall Windows sensors via the Carbon Black Cloud console.

Note You cannot uninstall Linux sensors via the Carbon Black Cloud console. You must uninstall Linux sensors by using the command line as explained in [Uninstall Linux Sensors from Workloads](#).

Procedure

- 1 Sign in to the Carbon Black Cloud console.
- 2 On the navigation bar, click **Inventory** and then click **Workloads**.
- 3 On the **Enabled** tab, select the sensors to uninstall.
- 4 Click **Take Action** and then click **Uninstall**. You are prompted to confirm the action.

Results

After you uninstall a sensor, it persists on the **Workloads** page as a deregistered sensor until you delete it. See [Delete Deregistered Sensors from Workloads](#)

Delete Deregistered Sensors from Workloads

You can delete deregistered sensors from Workloads manually or automatically.

Procedure

- ◆ To delete deregistered sensors from Workloads, use one of the following procedures.
 - **To manually delete deregistered sensors from workloads:**
 - a Sign in to the Carbon Black Cloud console.
 - b On the navigation bar, click **Inventory** and then click **Workloads**.
 - c On the **Enabled** tab, filter the list of sensors to show only devices that have deregistered sensors.
 - d Select the sensors to delete.
 - e Click **Take Action** and then click **Delete deregistered assets**. You are prompted to confirm the deletion.
 - **To automatically delete deregistered sensors from workloads:**
 - a Sign in to the Carbon Black Cloud console.
 - b On the navigation bar, click **Inventory** and then click **Workloads**.
 - c Click **Sensor Options** and then click **Manage Sensor Settings**.
 - d Select **Delete sensors that have been deregistered for** and set the time frame. Click **Save**.

Managing Kubernetes Sensors

10

You can secure your Kubernetes clusters and make them visible for Carbon Black Cloud by installing the Kubernetes Sensor on every cluster.

Before you start, make sure your Kubernetes environment meets the supported [Operating Environment Requirements for VMware Carbon Black Cloud Container Essentials](#).

If you want to review the clusters, which are already set up with the Kubernetes Sensor, see [Managing Kubernetes Clusters](#) in the *Carbon Black Cloud User Guide*.

This chapter includes the following topics:

- [Set Up the Kubernetes Sensor](#)
- [Upgrade the Kubernetes Sensor](#)
- [Edit a Kubernetes Cluster](#)
- [Delete a Kubernetes Cluster](#)
- [Kubernetes Cluster Status](#)

Set Up the Kubernetes Sensor

See how to set up and deploy the Carbon Black Cloud Kubernetes Sensor on your Kubernetes clusters.

The deployment and setup of the Kubernetes Sensor is performed with the help of a Kubernetes specific extension, called operator, along with an operator resource definition. Operators consist of set of controllers that deploy and manage components, defined by the user, and report on their health. The user defines the components with a custom resource definition.

The Carbon Black Cloud Operator deploys the Kubernetes Sensor inside the cluster and manages its lifecycle. The data in the custom resource file defines which features are enabled for the sensor. The essential steps of the sensor deployment procedure are:

- Setup and install the Carbon Black Cloud Operator
- Allow access to the Carbon Black Cloud console and
- Provide the Kubernetes Sensor configuration.

Procedure

- 1 On the left navigation pane, do one of the following depending on your system configuration and role:
 - If you have the Kubernetes Security DevOps or SecOps role and your system has only Containers Security feature, click **Inventory > Container Images**.
 - If you have any other role and your system has Containers Security and other Carbon Black Cloud features, click **Inventory > Kubernetes > Container Images**.

- 2 To add your Kubernetes cluster to the Carbon Black Cloud console, click **Add Cluster**.

The Add Cluster setup wizard appears.

- 3 On the **Cluster Detail** page, define the cluster that you are adding to the Carbon Black Cloud console.

| Attribute | Description |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Cluster name | Enter the name of the cluster. The cluster name must be unique, and it cannot contain a colon (:) symbol. |
| Cluster group | <p>You are able to create a high level grouping for your clusters, by defining cluster groups during the setup of the Kubernetes Sensor. By creating a cluster group, you can then specify a scope for your Kubernetes policies, spanning over that cluster group. The cluster group is also used for observing the network activity map of your clusters.</p> <ul style="list-style-type: none"> ■ Select an already existing cluster group from the list. ■ If you don't want to create or use cluster groups, enter default. |

- 4 On the **Authentication** page, you must provide a dedicated to the Kubernetes Sensor API key to establish the communication between your Kubernetes cluster and the Carbon Black Cloud console.

Do one of the following:

- Click to enable **Generate a new API key** and enter an API key name that is unique to your Carbon Black Cloud organization.
- Click to enable **Use existing API key** and select an existing API key.

Note We recommend that you do not reuse keys between clusters. Use a separate Carbon Black Cloud API key for each cluster.

- 5 On the **Sensor** page, make the following selections:
 - a Define the version of the Kubernetes Sensor to install on your cluster. The latest sensor version is set by default.
 - b Define the features you want to install, for example click to enable **Runtime protection** and **Cluster image scanning**.

- 6 On the **Finish Setup** page, execute consecutively the commands in the terminal of your Kubernetes environment. You can choose between Bash or PowerShell commands. Select **Bash** or **PowerShell** from the drop-down on the top right.

| Command | Description |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| First command | <p>The first command is about installing the Carbon Black Cloud Operator on your cluster, if it is not already installed, along with an operator resource definition. If the Carbon Black Cloud Operator is already installed, you can skip this command.</p> <p>To determine whether the operator is installed, in the terminal of your Kubernetes environment, run the command:</p> <pre>kubectl get pods -A -l control-plane=operator</pre> <p>If the Carbon Black Cloud Operator is there, you see the pod name and status.</p> <hr/> <p>Important The Carbon Black Cloud Operator resource definition might change between Kubernetes versions. The script automatically detects your Kubernetes version and determines the proper resource file to use.</p> |
| Second command | <p>The second command is about saving the API key as a Kubernetes secret in your cluster. Alternatively, you can add the secret to secrets management tool.</p> |
| Third command | <p>The third command is about installing the Kubernetes Sensor. You can alternatively use the YAML details as a command.</p> |

- 7 Click **Done**.

You see the cluster on the **Clusters** tab and the cluster status set to **Pending install**.

It takes up to 5 minutes for the cluster to stabilize during the initial setup.

During this time, the status might display as an error. We recommend waiting three to five minutes after submitting the install request to verify the correct status.

Results

After completing the setup procedure successfully, the status changes to **Running**.

Upgrade the Kubernetes Sensor

You may need to update the version of the Kubernetes Sensor, if you have an older version of the sensor installed on your clusters.

You can upgrade the Kubernetes Sensor either using the Carbon Black Cloud console or using a command-line interface.

To upgrade the Kubernetes sensor using a command-line interface, use the following command, where `value` is the latest version of the sensor. Note that the `cbcontainers-agent` is the sensor. For example, if the latest version of the sensor is 2.2.1, the command for upgrading the sensor to that version, is:

```
kubectl patch cbcontainersagent.operator.containers.carbonblack.io/cbcontainers-agent --
type=json' -p='[{"op": "replace", "path": "/spec/version", "value":"2.2.1"}]
```

To upgrade the sensor using the Carbon Black Cloud console, follow the procedure:

Procedure

- 1 On the left navigation pane, do one of the following depending on your system configuration and role:
 - If you are assigned Kubernetes Security DevOps role and your system has only Containers Security feature,
select **Inventory > Clusters**.
 - If you are assigned any other role and your system has Containers Security and other Carbon Black Cloud features,
select **Inventory > Kubernetes > Clusters**.
- 2 Find the cluster you want to update and in the **Actions** column, click the arrow and then click **Edit**.
- 3 Select the **Sensor version** from the list.
- 4 If there are features, not included with the previous installation, select each feature you want to include. For example, **Cluster image scanning**.
- 5 To run the upgrade, copy the command from the **Finish Setup** page, and run it in the terminal of your Kubernetes environment.

Edit a Kubernetes Cluster

You can edit a cluster in the Carbon Black Cloud console to enable features of the Kubernetes Sensor not included during the cluster setup.

Procedure

- 1 On the left navigation pane, do one of the following depending on your system configuration and role:
 - If you are assigned Kubernetes Security DevOps role and your system has only Containers Security feature,
select **Inventory > Clusters**.
 - If you are assigned any other role and your system has Containers Security and other Carbon Black Cloud features,
select **Inventory > Kubernetes > Clusters**.
- 2 Find the cluster you want to edit and in the **Actions** column, click the arrow and then click **Edit**.
- 3 Select the feature you want to include. For example, **Runtime protection** or **Cluster image scanning**.
- 4 To run the update, copy the command from the **Finish Setup** page, and run it in the terminal of your Kubernetes environment.

What to do next


For sensor upgrade, see [Upgrade the Kubernetes Sensor](#) procedure.

Delete a Kubernetes Cluster

You can remove a Kubernetes cluster from the Carbon Black Cloud console to stop observing it.

Procedure

- 1 On the left navigation pane, do one of the following depending on your system configuration and role:
 - If you are assigned Kubernetes Security DevOps role and your system has only Containers Security feature,
select **Inventory > Clusters**.
 - If you are assigned any other role and your system has Containers Security and other Carbon Black Cloud features,
select **Inventory > Kubernetes > Clusters**.

- 2 Find the cluster you want to remove from Carbon Black Cloud and click the icon  at the end of the row.

Delete Cluster window appears.

- 3 To delete the Kubernetes Sensor and the Carbon Black Cloud Operator from your cluster, copy the command from the **Delete Cluster** window, and run it in the terminal of your Kubernetes environment.

This step deletes the Kubernetes Sensor and the Carbon Black Cloud Operator from your cluster.

Important If you execute the command, without removing the cluster from the Carbon Black Cloud console in the next step, the cluster status becomes **Critical** after certain time. You can add the cluster again or remove it.

- 4 Click **Delete**.

This step removes the Kubernetes cluster from the Carbon Black Cloud console.

Important If you click Delete, without executing the command from the previous step, the Kubernetes Sensor and the Carbon Black Cloud Operator remain on your cluster without any activity on it. We recommend doing both - executing the command and removing the cluster.

Results

Deleting a cluster removes it from the Kubernetes clusters list on the **Clusters** tab.

Kubernetes Cluster Status

The Kubernetes cluster status indicates if the setup of the Kubernetes Sensor on your clusters is complete or if there are any warnings.

On the **Clusters** tab, the **Status** column indicates the status of a cluster.

- **Running:** All components are up and running without any errors.
- **Warning:** One of the non-critical components is down or cannot detect the status.
- **Error:** One of the critical components is down or cannot detect the status.
- **Critical:** No activity is detected from any cluster components for more than 24 hours.
- **Pending install:** Cluster setup is in progress.

Signature Mirror Instructions

11

This section contains Carbon Black Cloud signature mirror instructions for Linux and Windows.

Note The local scan feature is not available in the Audit and Remediation Standalone product.

See also [CB Defense: Getting Started with Local Mirror Servers](#).

This chapter includes the following topics:

- [Mirror Server Hardware Requirements](#)
- [Signature Mirror Instructions for Linux](#)
- [Signature Mirror Instructions for Windows](#)

Mirror Server Hardware Requirements

VMware Carbon Black Cloud mirror servers have the following hardware requirements to service 10,000 endpoints.

- 2Ghz CPU
- 4GB RAM

The recommended schedule for pulling down updates is hourly.

Performance of a local mirror server depends on the following:

- Number of endpoints that it serves
- Network bandwidth
- Frequency of updates

You can deploy multiple mirror servers to accommodate large environments.

Signature Mirror Instructions for Linux

This procedure provides instructions on mirroring a local Linux repository of the VMware Carbon Black Cloud local scanning signatures.

Assumptions

These instructions assume:

- A Linux operating system
- Definitions are hosted on an HTTP server at a given URL, which are entered in the **Update Servers** field of the **Local Scan** tab of a policy.

Procedure

- 1 Make sure that traffic to the signature update server URL is allowed without traffic inspection through any proxy/firewall (TCP/80 or TCP/443): `updates2.cdc.carbonblack.io`.
- 2 Download the `cbdefense_mirror_unix_x64_v3.0.zip` package from [CB Defense: Local Mirror Server for Signature Updates](#) to the server that will provide the updates.
- 3 Unpack the zipped file and move the contents to a directory. These files automate mirror server updates with a cron job, so they should be stored in a permanent location such as `/root/cbupdate`, `avupdate_msg.avr`, `avupdate.bin`, `HBEDV.KEY`, `update_defs.sh`, or `update_defs_ssl.sh`.
- 4 Open a command prompt window with administrative privileges and change the directory to the update file location.
- 5 Download the initial signature pack set and create the signature mirror by using the following command (`/var/www/html` is an example directory that is often used when configuring Apache): `bash ./update_defs.sh /var/www/html` The command can also call `update_defs_ssl.sh` to use https for the download.
- 6 Results print to the command line. Confirm that the following directories and files are located in the root of the directory that is targeted with the update command: `ave2`, `avupdate.log`, `idx`, and `x_vdf`.
- 7 Update the policy:
 - a Click **Enforce**, click **Policies**, and select the policy.
 - b Click the **Local Scan** tab.
 - c Enable **Allow Signature Updates**.
 - d Add the local mirror server URL to the **Update Servers** settings for internal and offsite devices.
 - e Check the box to the right of the URL to set it as primary.
 - f Click **Save**.

Results

For a detailed example of how to configure a signature mirror server on Apache, see [CB Defense: How to configure a Local Mirror \(Linux\)](#).

Signature Mirror Instructions for Windows

This procedure provides instructions on mirroring a local Windows repository of the VMware Carbon Black Cloud local scanning signatures.

Assumptions

These instructions assume:

- A 64-bit Windows operating system
- Definitions are hosted on an HTTP server at a given URL, which are entered in the **Update Servers** field of the **Local Scan** settings of a policy.

Procedure

- 1 Download the `cbdefense_mirror_win_x64_v3.0.zip` package from [CB Defense: Local Mirror Server for Signature Updates](#).
- 2 Extract `cbdefense_mirror_win_x64_v3.0.zip` files into a temp folder:
 - `avupdate.dll`
 - `do_update.bat`
 - `do_update_ssl.bat`
 - `HBEDV.KEY`
 - `msvcr120.dll`
 - `upd.exe`
 - `upd_msg.avr`
- 3 Create a folder for the AV signature update files; for example, `C:\inetpub\wwwroot\CBC_SignatureUpdates`.
- 4 Copy the extracted files into the folder that you created in Step 3.
- 5 Open `do_update.bat` and set `outdir` to the folder that you created in Step 3. (To use SSL, open `do_update_ssl.bat` instead.)
- 6 Configure the signature mirror by running the following commands in an elevated command prompt window: `C:\>cd C:\inetpub\wwwroot\CBC_SignatureUpdates`
`C:\inetpub\wwwroot\CBD_SignatureUpdates>do_update.bat`. The following folders are created:
 - `32`
 - `64`
 - `ave2`
 - `idx`

- x_vdf

7 Run Windows Task Scheduler.

- a Right-click **Task Scheduler Library** and click **Create Task**.
- b Click the **General** tab and define the task by adding a name and description. Select **Run whether user is logged on or not** and **Run with highest privileges**.
- c Click the **Triggers** tab. Click **New** and set the trigger to run daily at your preferred start time. Repeat the task every hour indefinitely. Select **Enabled** and click **OK**.
- d Click the **Actions** tab. Click **New** and **Start a program**. Set the **Program/script** to `do_update.bat`; for example, `C:\inetpub\wwwroot\CBC_SignatureUpdates>do_update.bat`. Click **OK**.
- e Click the **Conditions** tab. Select the following settings:
 - Start the task only if the computer is on AC power
 - Stop if the computer switches to battery power
 - Wake the computer to run this task
- f Click the **Settings** tab. Select the following settings:
 - Allow task to be run on demand
 - Run task as soon as possible after a scheduled start is missed
 - If the task fails, restart every > 1 minute
 - Attempt to restart up to > 3 times
 - If the running task does not end when requested, force it to stop
- g Click **OK**.

8 Create an IIS web site.

- a Open IIS Manager. Right-click **Sites** and click **Add Website**. Provide a site name that identifies that this web site is for the AV Signature Updates.
- b Keep the **DefaultAppPool** for the **Application Pool** field.
- c For the **Physical Path**, browse to the folder that was created in Step 3.
- d Keep these values: **Type** = **http**, **IP address** = **All Unassigned**, and **Port** = **80**.
- e For the **Host name** field, type the name of the mirror server.
- f Select **Start Website immediately**. Click **OK**.
- g On the IIS navigation pane, under **Sites**, select the site name that you created in Step 8a.
- h Double-click **Directory Browsing** and click **Enable**
- i Double-click **MIME Types**. Add a new MIME type for extension of `.idx` with type of **text/plain**.

- j In a command prompt window with administrative privileges, run the command `iisreset`.
 - k To test the URL, open a browser and type **http://{ host name from Step 8e}**. You should see the folders that were created in Step 6.
- 9 Update the policy.
- a In the Carbon Black Cloud console, click **Enforce**, click **Policies**, and select the policy.
 - b Click the **Local Scan** tab and enable **Allow Signature Updates**.
 - c Add the local mirror server URL to the **Update Servers** settings for internal and offsite devices. Check the box to the right of the URL to set it as the primary. Click **Save**.

Results

Note

- `do_update.bat` generates (and appends to) a log file in `%TEMP%\scanner\upd.log`. You can use this log file to troubleshoot issues.
 - The **Update Servers for Onsite Devices** checkbox on the **Local Scan** tab in a policy can impact connections to the mirror server. If you have sensors that can't receive updated signatures from the mirror server, toggle the switch to resolve the issue.
-

Configuring Carbon Black Cloud Communications

12

Configure your network infrastructure and endpoints to ensure proper communication between sensors and the backend.

The current implementation of the Carbon Black Cloud service uses dynamically managed load balancers to provide the best possible levels of scalability, reliability, and performance. The Carbon Black Cloud services hostname resolves several possible IP addresses that can change dynamically.

There is no static IP, range of IP addresses, or subnet to allow or exclude in firewall or proxy settings.

Network proxies and firewalls can interfere with communication between the Carbon Black Cloud sensor and the Carbon Black Cloud backend if they are improperly configured.

This chapter includes the following topics:

- [Configure a Firewall](#)
- [Configure a Proxy](#)

Configure a Firewall

A sensor can connect to the backend in a firewall-protected network in several ways.

URLs are used for the following purposes:

- Console/API — Console access and API requests
- Sensor — Communication between the sensor and the console/backend
- UBS download — Downloading Unified Binary Store (UBS) binaries and metadata
- Content management — UBS and dynamic rules engine updates
- Signature — Updating signature packs
- Third-party certificate validation — Verifying sensor comm certificates
- Live Response Uploads - Used when performing the "get" command from Live Response

Configure the firewall to allow incoming and outgoing TCP/443 (default) and TCP/54443 (backup) connections to the following environment specific URLs:

Table 12-1. Environment-specific URLs

| Environment/AWS Region | Console/API | Sensor | UBS download | Live Response Uploads |
|--------------------------|---------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| Prod01 (US-East-1) | https://dashboard.confer.net | https://devices.confer.net | https://cdc-file-storage-production-us-east-1.s3.amazonaws.com | https://defense-cblr-file-uploads-us-east-1.s3.amazonaws.com |
| Prod02 (US-East-1) | https://defense.conferdeploy.net | https://dev5.conferdeploy.net | https://cdc-file-storage-production-us-east-1.s3.amazonaws.com | https://defense-cblr-file-uploads-us-east-1.s3.amazonaws.com |
| Prod05 (US-East-1) | https://defense-prod05.conferdeploy.net | https://dev-prod05.conferdeploy.net | https://cdc-file-storage-production-us-east-1.s3.amazonaws.com | https://defense-cblr-file-uploads-us-east-1.s3.amazonaws.com |
| Prod06 (EU-Central-1) | https://defense-eu.conferdeploy.net | https://dev-prod06.conferdeploy.net | https://cdc-file-storage-production-eu-central-1.s3.amazonaws.com | https://defense-cblr-file-uploads-eu-central-1.s3.amazonaws.com |
| ProdNRT (AP-Northwest-1) | https://defense-prodnrt.conferdeploy.net | https://dev-prodnrt.conferdeploy.net | https://cdc-file-storage-production-ap-northeast-1.s3.amazonaws.com | https://defense-cblr-file-uploads-ap-northeast-1.s3.amazonaws.com |
| ProdSYD (AP-Southwest-2) | https://defense-prodsyd.conferdeploy.net/ | https://dev-prodsyd.conferdeploy.net/ | https://cdc-file-storage-production-ap-southeast-2.s3.amazonaws.com | https://defense-cblr-file-uploads-ap-southeast-2.s3.amazonaws.com |

Additionally, all environments use the following URLs:

Table 12-2. All environments

| Category | URL | Protocol/Port | Notes |
|------------------------|-------------------------------------------------------------------------------------------------------|---------------|--------------------------------------|
| Content Management URL | https://content.carbonblack.io | TCP/443 | |
| Signature URL | http://updates2.cdc.carbonblack.io/update2 | TCP/80 | Windows sensor versions prior to 3.3 |
| Signature URL | https://updates2.cdc.carbonblack.io/update2 | TCP/443 | Windows sensor versions 3.3+ |

Table 12-2. All environments (continued)

| Category | URL | Protocol/Port | Notes |
|----------------------------------------|-------------------------|---------------|--------------------------------------------------------------------------------------------------------------------------|
| Third-party certificate validation URL | http://ocsp.godaddy.com | TCP/80 | Online Certificate Status Protocol (OCSP). Sensor version 3.3+: required unless <code>CURL_CRL_CHECK</code> is disabled. |
| Third-party certificate validation URL | http://crl.godaddy.com | TCP/80 | Certificate Revocation List (CRL). Sensor version 3.3+: required unless <code>CURL_CRL_CHECK</code> is disabled. |

If you do not make specific network firewall changes to access the Carbon Black Cloud backend applications, the sensors try to connect through existing proxies. See [Configure a Proxy](#).

Note Operational environments that implement a man-in-the-middle proxy should note that additional third-party certificate validation URLs can be needed depending on the server certificates that the proxy uses. Additional URLs include anything specified under the "CRL Distribution Points" and "Authority Information Access" extensions of the proxy server SSL certificate. Failing to allow communication to third-party certificate validation URLs on TCP port 80 can lead to communication failures between the sensor and the backend. The Windows 3.3 and higher sensor relies on Windows to execute a CRL check. This sensor communication certificate verification is recommended but not required. If the sensor fails to validate its own communication certificate, installation will fail unless you set `CURL_CRL_CHECK=0` (see [Disable CURL CRL CHECK](#)).

If installation fails for this reason and you do not want to disable the CRL check, you can implement one of the following options:

- Configure the Winhttp service to use the proxy for Windows CRL checks
- Configure the proxy or firewall to allow CRL traffic
- Allow port 80 traffic to `crl.godaddy.com` and `ocsp.godaddy.com` through the proxy or firewall

Carbon Black Cloud Workload Appliance

| Carbon Black Service URL / Hostname | IP Address | Protocol/Port | Description |
|--------------------------------------------------------------------------------------------------------------------------------------------|--------------|---------------|--------------------------------------------|
| prod.cwp.carbonblack.io | Dynamic | TCP/443 | Appliance logging and updates. |
| vCenter Server Host | User defined | TCP/443 | Communication with the vCenter Server . |
| Carbon Black Cloud console URL (refer to Console/API URL) For example, https://defense-prod05.conferdeploy.net if you are a Prod05 user | Dynamic | TCP/443 | Communication with the Carbon Black Cloud. |

Disable CURL CRL CHECK

The `crl.godaddy.com` and `ocsp.godaddy.com` domains use OCSP (Online Certificate Status Protocol) and Certificate Revocation List (CRL) checks to validate a sensor's install certificate. You can disable this check.

Prerequisites

Caution You can disable CRL checks either during or after a sensor installation. However, disabling CRL can potentially open devices up to *man in the middle* attacks if Carbon Black Cloud revokes the certificate (this has never happened), and if an attacker then leverages the revoked certificate for such an attack.

To disable CRL check during an initial sensor install:

Using the command line install method, add the `CURL_CRL_CHECK=0` option to the install command. For example:

```
msiexec.exe /q /i CBDefense-setup.msi /L*vx log.txt CURL_CRL_CHECK=0
```

To disable CRL checks after the sensor is installed:

Procedure

- 1 In the Carbon Black Cloud console, click **Inventory** and then click **Endpoints**.
- 2 Select the endpoint, click **Take Action**, and then click **Enable bypass**. Confirm the action.
- 3 To confirm that the endpoint is in bypass mode, run the following RepCLI command: `repcli status`

- 4 As a best practice, create a backup of the `cfg.ini` file into another directory.
For Windows sensor versions 3.6 and earlier, `cfg.ini` is located at `C:\Program Files\Confer\cfg.ini`. For Windows sensors 3.7 and later, `cfg.ini` is located at `C:\ProgramData\CarbonBlack\DataFiles\cfg.ini`. After you successfully complete the procedure, delete the backup file.
- 5 Edit `cfg.ini`. Add the following parameter to the end of the file: `CurlCr1Check=false`.
- 6 Run the following RepCLI command: `RepCLI updateconfig`.
- 7 In the Carbon Black Cloud console, click **Inventory** and then click **Endpoints**.
- 8 Select the endpoint, click **Take Action**, and then click **Disable bypass**. Confirm the action.
See also [Configure a Firewall](#). For more information about RepCLI, see [Managing Sensors by using RepCLI](#) in the User Guide.

Configure a Proxy

The Carbon Black Cloud sensor uses a variety of mechanisms to determine whether a network proxy is present.

If a proxy is detected (or if one is specified at install time), the sensor attempts to use that proxy. If no proxy is detected, the sensor will attempt a direct connection through port 443 or 54443.

The sensor attempts to contact the Carbon Black Cloud backend by using the following methods:

- A static configured proxy that is configured during sensor installation.
- A direct connection over TCP/443.
- Auto-detection of a proxy and proxy credentials (when applicable) from the local computer's operating system settings.

If you cannot establish connectivity over the standard SSL port, the sensor can fail over to the alternate port, which is TCP/54443.

Note Carbon Black Cloud sensors automatically try to detect proxy settings during initial installation. This should be tested. If the automatic proxy detection doesn't succeed, you must define the parameters to include the Proxy IP and Port in the MSI command line during a command line installation.

If user authentication is required, the user might be prompted for credentials. This typically does not occur in environments that require proxy credentials because the sensor uses an existing configuration that avoids requiring end users to enter credentials.

Note Windows sensor 3.3 and later versions performs a CRL check. OCSP and CRL traffic is not handled directly by the sensor or the installer, and does not use the proxy parameters that are specified at install. This traffic requires having WinHTTP set to the proxy.

You must either disable the CRL check (see [Disable CURL CRL CHECK](#)), or configure WinHTTP to use an existing `proxy server:port`. You can perform the latter option in the following ways:

- Set WinHTTP proxy information through proxy-side configuration.
- Manually set WinHTTP proxy through a command line interface on specific machines:

```
netsh winhttp set proxy <proxy>:<port>
```

- Set WinHTTP on multiple machines by using Group Policy.

To avoid going through a network proxy (and/or to avoid being blocked by a firewall), you might need to configure a bypass on your proxy server/firewall to allow outgoing connections from the sensor to the backend. Options for bypass configuration include the following:

- Configure a bypass on your firewall or proxy to allow outgoing connections to your Carbon Black Cloud domain over TCP/443.
- Configure a bypass in your firewall or proxy to allow outgoing connections to the Carbon Black Cloud alternate port TCP/54443.

Important The host domain name for the Carbon Black Cloud backend server is included in the server's certificate. Some network proxies and gateways might try to validate the certificate and deny the Carbon Black Cloud backend application connection because of a name mismatch between the certificate and real host name of the system that is running in AWS. If this occurs, you must configure the proxy or gateway so that it does not validate the backend server certificate. Note that you cannot access the certificate or hostname in the server's certificate.

Connection Mechanism Precedence

If a sensor fails to connect to the backend, it tries the last known working settings, starting with the most recent ones.

These include the following:

- Proxy
- No proxy
- Credentials
- No credentials
- Proxy used at install time

- Direct connection
- Alternate 54443 port

If the sensor cannot connect using its last valid settings, it reattempts the connection in the following sequence:

- 1 The proxy server that was provided during sensor installation (if applicable).
- 2 Variants of the proxy that was set during sensor installation (if applicable). These variants are with or without credentials using default port (443) and the alternate port (54443).
- 3 A direct connection to the backend with no proxy and default port (443).
- 4 A direct connection to the backend with no proxy using the alternate port 54443.
- 5 Dynamically set proxies such as:
 - Proxies configured within `inetcp1.cpl` (Internet Options) – For each server, also try default (443) and alternate (54443) ports.
 - `.pac` files configured within `inetcp1.cpl` – For each server, also try default (443) and alternate (54443) ports.

Note Sequence numbers 1 and 2 can be switched by using `PreferStaticProxyOverLastUsed=true` as described in [Configure a Proxy for Windows after Sensor Installation](#).

Configure a Proxy for Windows after Sensor Installation

This article describes how to configure a proxy for Windows after the sensor has been installed.

Prerequisites

This procedure requires that you have RepCLI authentication. For more information about RepCLI, see [Managing Sensors by using RepCLI](#) in the User Guide.

Procedure

- 1 Place the sensor into bypass mode:

```
repcli bypass 1
```

- 2 Confirm that the sensor is in bypass mode:

```
repcli status
```

- 3 Shut down the sensor service:

```
repcli stopCbServices
```

- 4 As a best practice, create a backup of the `cfg.ini` file into another directory.
For Windows sensor versions 3.6 and earlier, `cfg.ini` is located at `C:\Program Files\Confer\cfg.ini`. For Windows sensors 3.7 and later, `cfg.ini` is located at `C:\ProgramData\CarbonBlack\DataFiles\cfg.ini`. After you successfully complete the procedure, delete the backup file.

- 5 Edit `cfg.ini` in a plain text editor.

- a If the following parameters exists in `cfg.ini`, remove them:

```
ProxyServer=
ProxyServerCredentials=
```

- b Add the following parameters:

```
ProxyServer=[PROXY_IP_OR_DOMAIN]:[PROXY_PORT]
ProxyServerCredentials=[USERNAME]:[PASSWORD] (Optional- if proxy requires
authentication)

---Example---
ProxyServer=TestProxy.net:8080
ProxyServerCredentials=TestUsername:TestPassword
```

- c If the original proxy is still functioning, add the following value to override the previously used value. This option is only available in Windows sensors 3.6+.

```
PreferStaticProxyOverLastUsed=true
```

- 6 Save `cfg.ini`.

- 7 Restart the sensor service:

```
sc start CbDefense
```

- 8 Take the sensor out of bypass mode:

```
repcli bypass 0
```

- 9 To force an immediate check-in (optional):

```
repcli cloud hello
```

Configure a Proxy for Linux (all Sensor Versions)

Use this procedure to configure a proxy through the `cfg.ini` file for all distributions.

Procedure

- 1 Extract the contents of the installer package into a temporary directory.

- 2 Use the `install.sh` script to install the agent, but do not provide a company code:

```
sudo cb-psc-install/install.sh
```

- 3 Update the `cfg.ini` file with the v3.x+ company code:

```
sudo /opt/carbonblack/psc/bin/cbagentd -d '<COMPANY_CODE>'
```

- 4 Append the following entry in the `/var/opt/carbonblack/psc/cfg.inifile`. You can use the IP address instead of the hostname.

```
ProxyServer=<hostname>:<port number>
```

Note The Linux sensor only supports a HTTP non-authenticated proxy server through `cfg.ini`.

Table 12-3.

| Proxy Type | IP Format | FDQN Format |
|------------|----------------|------------------|
| HTTP | ip:port | fdqn:port |
| HTTP | http://ip:port | http://fdqn:port |

Example `Cfg.ini` settings:

```
[customer]
ProxyServer=proxy.example.com:3128
```

ProxyServer=<hostname>:<port number>

- 5 Start the agent:
 - Centos/Rhel 6:
 - `$ service cbagentd start`
 - All other distributions:
 - `$ systemctl start cbagentd`

Configure a Proxy for Linux (Sensor Versions 2.11.1+)

For Linux sensor version 2.11.1 onwards, use this procedure to configure a proxy through the `install.sh` script for all distributions.

Procedure

- 1 Extract the contents of the installer package into a temporary directory.

- 2 Use the `install.sh` script to install the agent together with proxy server details and the company code.

You can use the IP address or hostname as part of `ProxyHost`.

```
sudo cb-psc-install/install.sh -p 'ProxyHost:ProxyPort' '<COMPANY_CODE>'
```

macOS Proxy Server Information

The macOS 3.7.2 sensor uses macOS Keychain APIs to improve proxy server information storage.

If you downgrade the macOS sensor from 3.7.2+ to an earlier sensor version, proxy settings must be repopulated.

- If you used macOS System Preferences for the proxy configuration, the sensor attempts to repopulate the proxy information after the sensor is downgraded.
- If the proxy configuration cannot be retrieved from the macOS System Preferences, you must use the sensor unattended installer options `-p PROXY_SERVER:PORT` and `-x PROXY_USER:PASSWORD` to repopulate the proxy settings during the sensor downgrade.