Deploy Falcon Sensor for Linux Using CLI

Last updated: Nov. 14, 2023

Overview

Deploying Falcon sensor for Linux on the host provides runtime protection for the host and, if applicable, the containers running on the host.

To install the Falcon sensor on an endpoint or node, perform these procedures:

1. Install the Falcon sensor
2. Verify the installation
3. Optionally, configure the Falcon sensor with post-installation options

For information about other installation considerations, see
Installation Options for Falcon Sensor for Linux [/documentation/page/f4d593ca/installation-options-for-falcon-sensor-for-linux].

Before you begin

Before you install the Falcon sensor for Linux, refer to these topics to make sure you are installing the correct sensor, your environment meets system requirements, you are using the correct deployment method and installation type, you are aware of any additional installation options your installation requires:

- Choosing the right sensor [/documentation/page/fab6a179/choose-the-right-sensor]
- Falcon sensor for Linux system requirements [/documentation/page/edd7717e/falcon-sensor-for-linux-system-requirements]
- Planning your Falcon sensor for Linux deployment [/documentation/page/e99b9f4f/plan-your-falcon-sensor-for-linux-deployment]
- Installation Options for Falcon Sensor for Linux [/documentation/page/f4d593ca/installation-options-for-falcon-sensor-for-linux]

Installing the Falcon sensor

1. Download the Falcon sensor installer from Host setup and management > Deploy > Sensor Downloads [/hosts/sensor-downloads].
2. Copy your Customer ID Checksum (CID), displayed on Sensor Downloads.
3. Run the installer, substituting `<installer_filename>` with your installer's file name.
   Installing the sensor requires sudo privileges.
   - Ubuntu: `sudo dpkg -i <installer_filename>`
   - RHEL, CentOS, Amazon Linux: `sudo yum install <installer_filename>`
   - SLES: `sudo zypper install <installer_filename>`
4. Set your CID on the sensor, substituting <CID> with your CID.
   - All OSes: `sudo /opt/CrowdStrike/falconctl -s --cid=<CID>`
5. Start the sensor manually.
   - Hosts with SysVinit: `sudo service falcon-sensor start`
   - Hosts with Systemd: `sudo systemctl start falcon-sensor`

Verifying sensor installation

You can verify the sensor installation by using the Falcon console or a terminal on the host:

Falcon console	Host

After the sensor is installed, the host connects to the CrowdStrike cloud. You can confirm a sensor installation by reviewing your hosts in the Falcon console.
Search for the host in **Host setup and management > Manage endpoints > Host management**.
To view a complete list of newly installed sensors, use the Sensor Report [/investigate/events/en-US/app/eam2/sensor_app].
	To validate that the Falcon sensor for Linux is running on a host, run this command at a terminal:
	`ps -e | grep falcon-sensor`
	You'll see output similar to this:
	`[root@centos6-installtest ~]# sudo ps -e | grep falcon-sensor`
	`905 ? 00:00:02 falcon-sensor`

Verifying user mode

For hosts on Falcon sensor for Linux version 6.51 and later, go to the Host management page to see hosts currently in user mode. See Host and Host Group Management [/documentation/page/f8a0f751/host-and-host-group-management].

Additionally, run one of the following queries to see Linux hosts online within the last 24 hours that report being in user mode.

> **Note:** The Raptor release, which includes the CrowdStrike Query Language (CQL), is being rolled out in waves. When your wave begins, you'll gain access to CQL. Until then, use the legacy query language syntax. For more info, see Raptor Release Resource Center [/documentation/page/de2919a2/raptor-release-resource-center].

- CrowdStrike Query Language syntax
  Run this query:

```
#event_simpleName=/(OsVersionInfo|ConfigStateUpdate|SensorHeartbeat)/
| event_platform=Lin
| groupBy([cid, aid], function=([selectLast([ConfigStateData, OSVersionString, SensorStateBitMap])]))
| ConfigStateData match {
/1400000000c4/ => isInUserMode:="1" ;
* => isInUserMode:="0" ;
}
| SensorStateBitMap match {
2 => isInRFM:="1" ;
0 => isInRFM:="0" ;
}
| case {
isInUserMode=1 AND isInRFM=1 | message := "User Mode Enabled" ;
isInUserMode=0 AND isInRFM=0 | message := "Kernel Mode Enabled" ;
isInUserMode=0 AND isInRFM=1 | message := "RFM" ;
* }
| OSVersionString = /^Linux\s(?<hostName>\S+)\s(?<kernelVersion>\S+)\s/i
| aid=?aid hostName=?hostName message=?message
| select([aid, hostName, Version, kernelVersion, isInRFM, isInUserMode, message])
```

- Legacy query language syntax
  Run this query:

```
earliest=-26h event_platform=Lin event_simpleName IN (ConfigStateUpdate, SensorHeartbeat, OsVersionInfo) | stats latest(ConfigStateData) as ConfigStateData, latest(SensorStateBitMap_decimal) as SensorStateBitMap_decimal, latest(OSVersionString) as OSVersionString by cid, aid | rex field=OSVersionString "Linux\\s\\S+\\s(?<kernelVersion>\\S+)?\\s.*" | eval ConfigStateData=split(ConfigStateData, ",") | eval userModeEnabled=if(match(ConfigStateData,"1400000000c4"),"Yes","No") | eval rfmFlag=if(match(SensorStateBitMap_decimal,"2"),"Yes","No") | eval sensorState=case( userModeEnabled == "Yes" AND rfmFlag == "Yes", "User Mode Enabled", userModeEnabled == "No" AND rfmFlag == "No", "Kernel Mode Enabled", userModeEnabled == "No" AND rfmFlag == "Yes", "RFM", true(),"-") | lookup local=true aid_master.csv aid OUTPUT ComputerName, AgentVersion as falconVersion, Version as osVersion, FirstSeen, Time as LastSeen | fillnull kernelVersion value="-" | table aid, ComputerName, falconVersion, osVersion, kernelVersion, sensorState, osVersion, FirstSeen, LastSeen | convert ctime(FirstSeen) ctime(LastSeen) | sort + ComputerName
```

Installation Options for Falcon Sensor for Linux > [/documentation/page/f4d593ca/installation-options-for-falcon-sensor-for-linux]