Granular status dashboards to identify Windows hosts impacted by content issue (v8.6)

Published Date: Jul 22, 2024

Objective

- Identify Windows hosts impacted by the content update defect in this Tech Alert

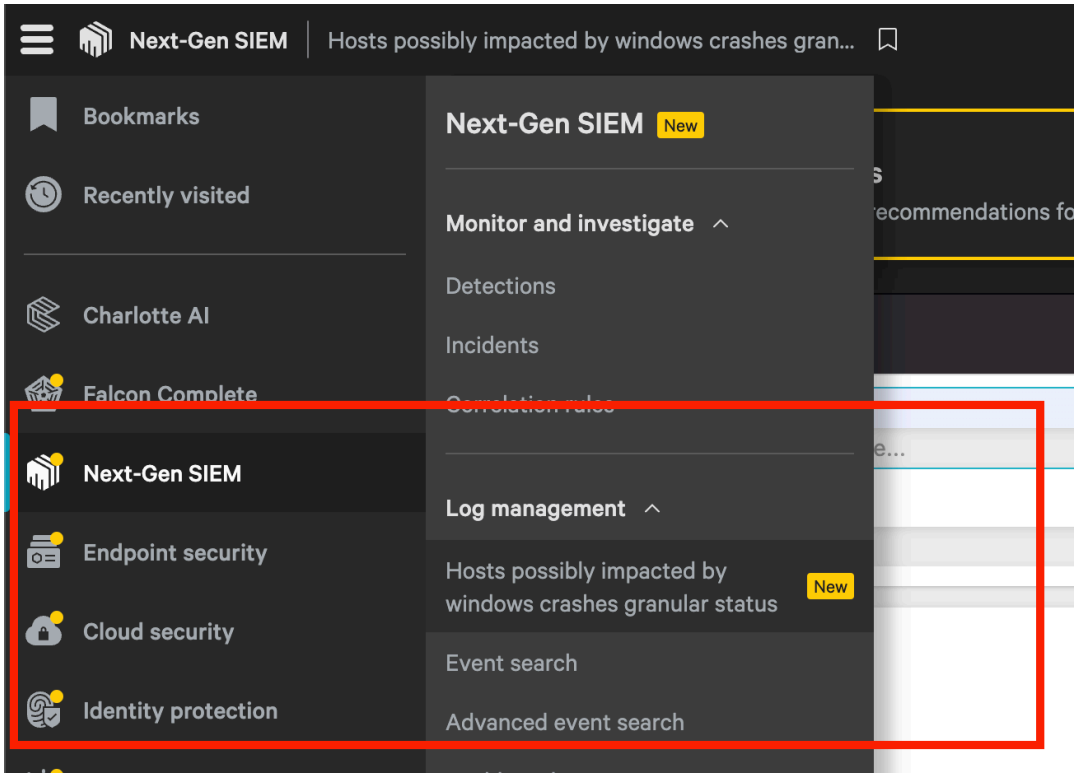
Applies To

- Supported versions of the Falcons sensor for Windows
- Supported versions of Microsoft Windows
- Falcon Insight XDR subscription
- Related to Tech Alert | Windows crashes related to Falcon Sensor | 2024-07-19

Procedure

An updated granular dashboard is available that displays the Windows hosts impacted by the content update defect described in this Tech Alert.

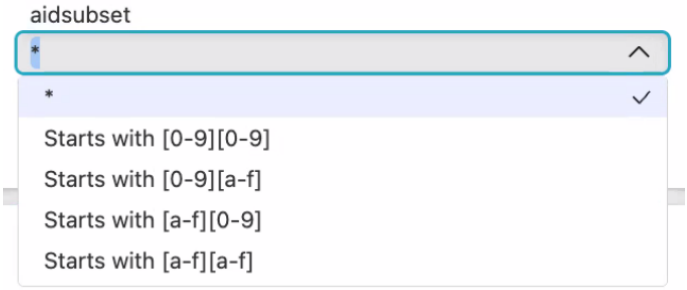
- Find the updated dashboard at: **Next-Gen SIEM > Log management > Hosts possibly impacted by windows crashes granular status**



- Note: The Dashboard cannot be used with the "Live" button

On the dashboard, you must select values for these two fields:

1. In the **cid** field, select a CID in your environment.
    a. If you have multiple CIDs (Falcon Flight Control) and the dashboard does not return any data, enter an asterisk (*) in the CID field to get data from all of the CIDs in your environment.
2. In the **aidsubset** field, select at least one group or the asterisk (*).
    a. The **aidsubset** field contains a drop-down menu to allow organizations with large amounts of hosts (AIDs) to filter the data results into smaller subsets. The filters are based on the first two characters of the host's AID and are presented as regular expressions:

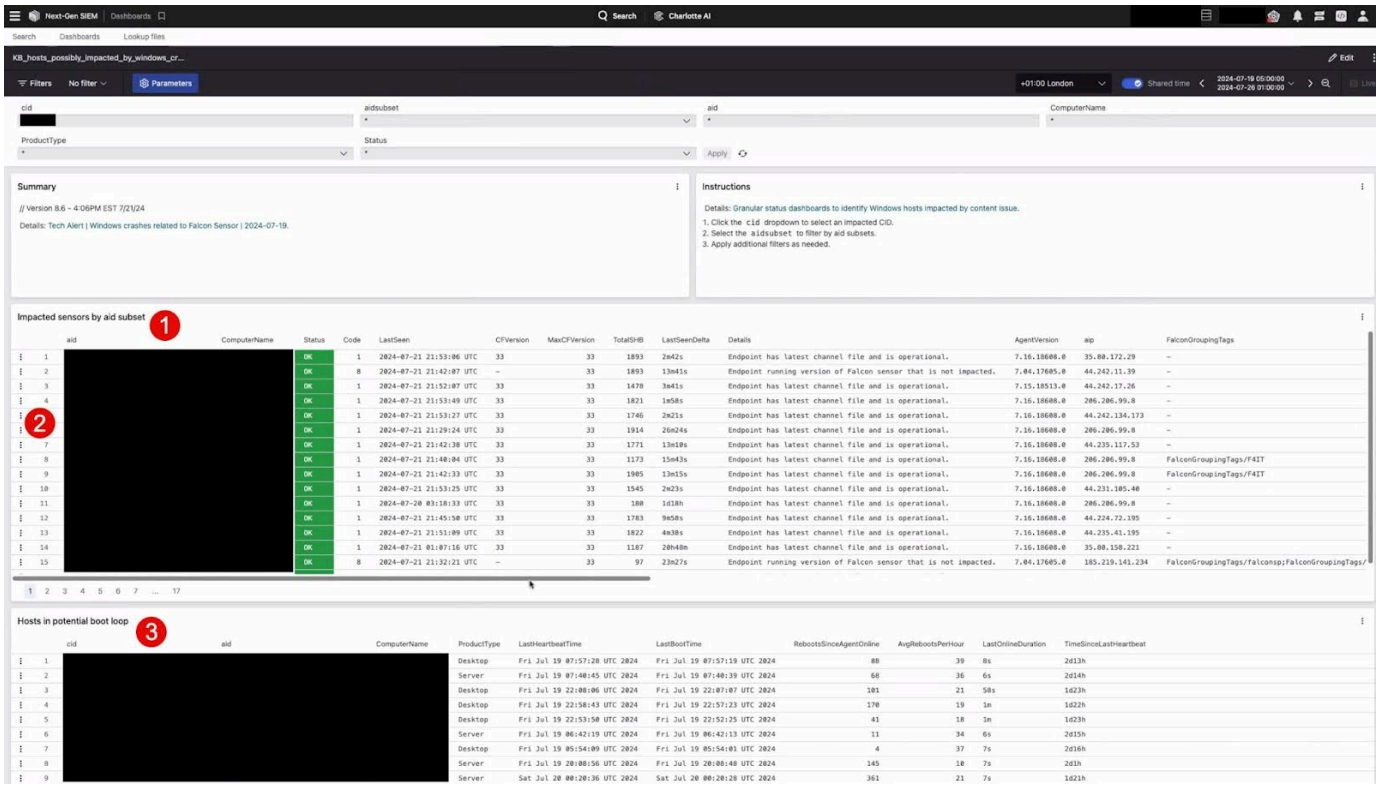


Optionally, filter by additional details:

1. In the **Status** field, leave the asterisk (*) unchanged to see all host statuses or select a specific value to see hosts with that status.
2. In the **ComputerName** field, leave the asterisk (*) unchanged to see all hosts or enter a computer name to see a specific host.
3. Click **Apply** to reload the dashboard.

Granular status dashboard widgets overview

1. **Impacted sensors by aid subset** (labeled 1): Shows the status of the host selection group based on the AID subset provided.
2. **3-dot menu** (labeled 2): Select **See host details** from the 3-dot menu to go to the **Drill-down dashboard**.
3. **Hosts in potential boot loop** (labeled 3): Shows systems that are actively stuck in a boot loop and have an AID that's reporting to Falcon as unhealthy.
    - NOTE: If a sensor is uninstalled, the AID remains on this list as it continues reporting into Falcon as unhealthy.



Impacted sensors by aid subset

The **Impacted sensors by aid subset widget** uses data derived from **ConfigStateUpdate** and **SensorHeartbeat** events. Using this data, the widget classifies hosts with **Status** and **Details** that indicate the likely current state of each host.

NOTE: hosts in a boot loop may be unable to reliably deliver telemetry to the CrowdStrike cloud. This may impact the assessments made by the dashboard. The assessments displayed in the dashboard may change over time as hosts fully recover.

This widget uses the **ConfigStateUpdate** data to provide the host status of **DOWN, VERIFY, RECOVERY_LIKELY, RECOVERY_VERY_LIKELY, UNKNOWN**, or **OK** (see **Status definitions and details**, below). Process the dashboard findings in this priority order.

Status definitions and details

Each status is based on available data and includes possible details.

- **DOWN:** a high confidence assessment where remediation is likely to be required
    - Endpoint has channel file version of 0 and has not checked-in after impact window.
    - Endpoint received channel file during impacted window, but endpoint has NOT checked-in after impact window.
- **VERIFY:** a low to medium confidence assessment

    - Endpoint received channel file during impact window and has checked-in after impact window.
- **RECOVERY_LIKELY:** a medium confidence assessment
    - Endpoint received channel file during impact window and has checked-in after impact window with a total reported uptime of 5-10 hours.
- **RECOVERY_VERY_LIKELY:** a medium to high confidence assessment
    - Endpoint received channel file during impact window and has checked-in after impact window a total reported uptime of 10-20 hours.
- **UNKNOWN:** there is not enough available data to form an assessment
    - Cannot determine endpoint status based on available telemetry.
- **OK:** a high confidence assessment
    - Endpoint running version of Falcon sensor that is not impacted.
    - Endpoint has latest channel file and is operational.
    - Endpoint was offline and did not receive channel file during impact window.
    - Endpoint was online and did not receive channel file during impact window.

Cross-reference other columns in the widget to determine a more accurate status of each host. For example:

- The **LastSeen** time of the host (and/or the **LastSeenDelta** field)
- Whether or not the **CFVersion** column matches the **MaxCFVersion** column
    - **CFVersion** is the channel file version observed on the host. **MaxCFVersion** is the most recent version of the channel file on the CID
- The total number of **SensorHeartbeat** events received after time of impact, as indicated by the **TotalSHB** column
- The **ProductType** (Desktop, Server, or Domain Controller)
- Additional contextual data, including **Tags** (FalconGroupingTags or SensorGroupingTags), **MachineDomain, SiteName, OU, LocalAddressIP4,** and **MAC**

NOTE: If you are unsure of any status value (including systems in the VERIFY or UNKNOWN status), use the drill-down dashboard to investigate further.

Hosts in potential boot loop

The **Hosts in potential boot loop** widget uses data derived from **AgentOnline** and **SensorHeartbeat** events. **AgentOnline** events are sent by the sensor on restart (for example, host reboot, sensor upgrade or downgrade). A properly functioning sensor sends **SensorHeartbeat** events every two minutes.

Using this data, the table displays hosts with:

- Sensor version 7.11.18110 or later (earlier sensor versions are not impacted by the content update defect)
- Most recent **Uptime** of 10 minutes or fewer

    - **Uptime** is derived from the timestamp of the most recent AgentOnline event and the most recent SensorHeartbeat event

The **Hosts in potential boot loop** widget examines AgentOnline events and has these columns:

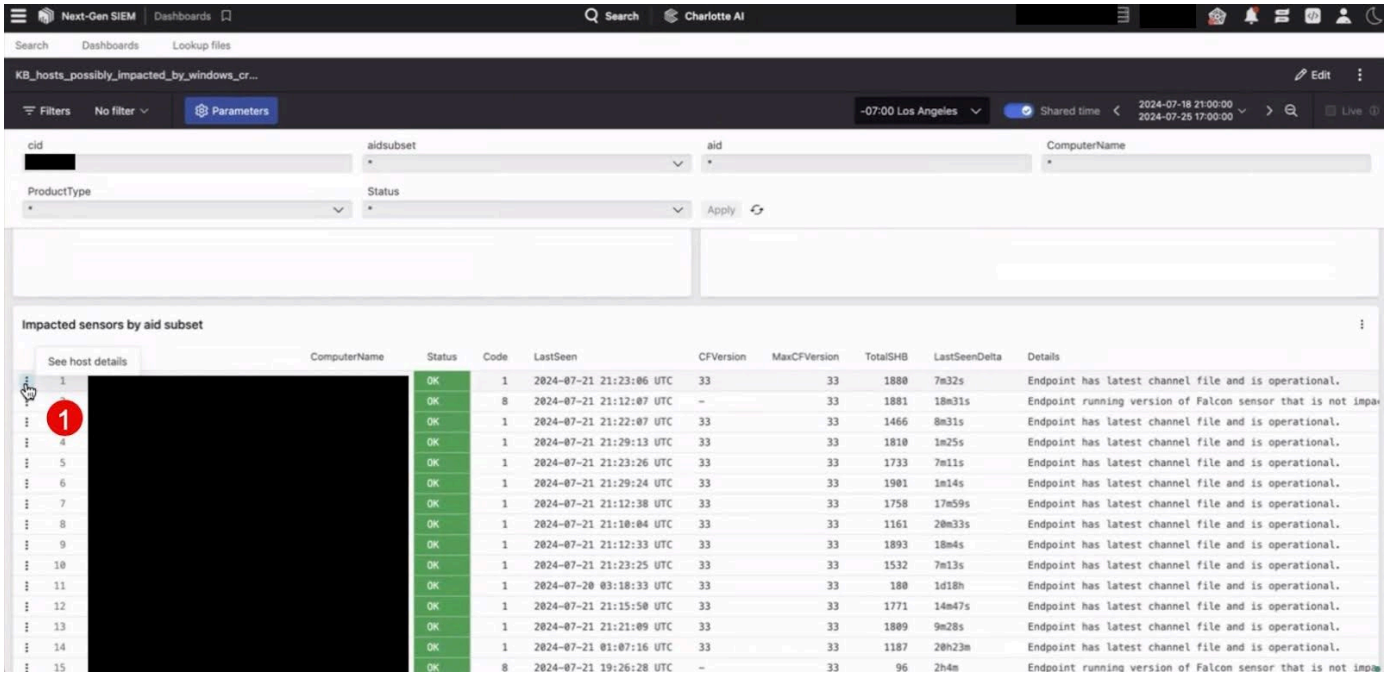
- **cid:** The Customer ID
- **Aid:** The Agent ID (or the host ID)
- **ComputerName:** The hostname
- **ProductType:** Desktop, Server, or Domain Controller
- **LastHeartbeatTime:** The timestamp of the most recent **SensorHeartbeat** event
- **LastBootTime:** The timestamp of the most recent **AgentOnline** event
- **RebootsSinceAgentOnline:** The delta between the minimum observed **boot identifier** value and the maximum observed **boot identifier** value, where boot identifier is the number of reboots made since Windows was first installed on the host
- **AvgRebootsPerHour:** Hourly average of R**ebootsSinceAgentOnline**
- **LastOnlineDuration:** The delta between **LastHeartbeatTime** and **LastBootTime**
- **TimeSinceLastHeartbeat:** The delta between the current time and **LastHeartbeatTimer**

Use case ideas

- An organization that has a downstream impact of automated tickets should focus on reducing the **AvgRebootsPerHour** number. Remediating these hosts first allows the downstream impact to be reduced.
- An organization that focuses on reducing the total downtime for a given host should sort by **TimeSinceLastHeartbeat** to prioritize hosts that have been down for an extended period.

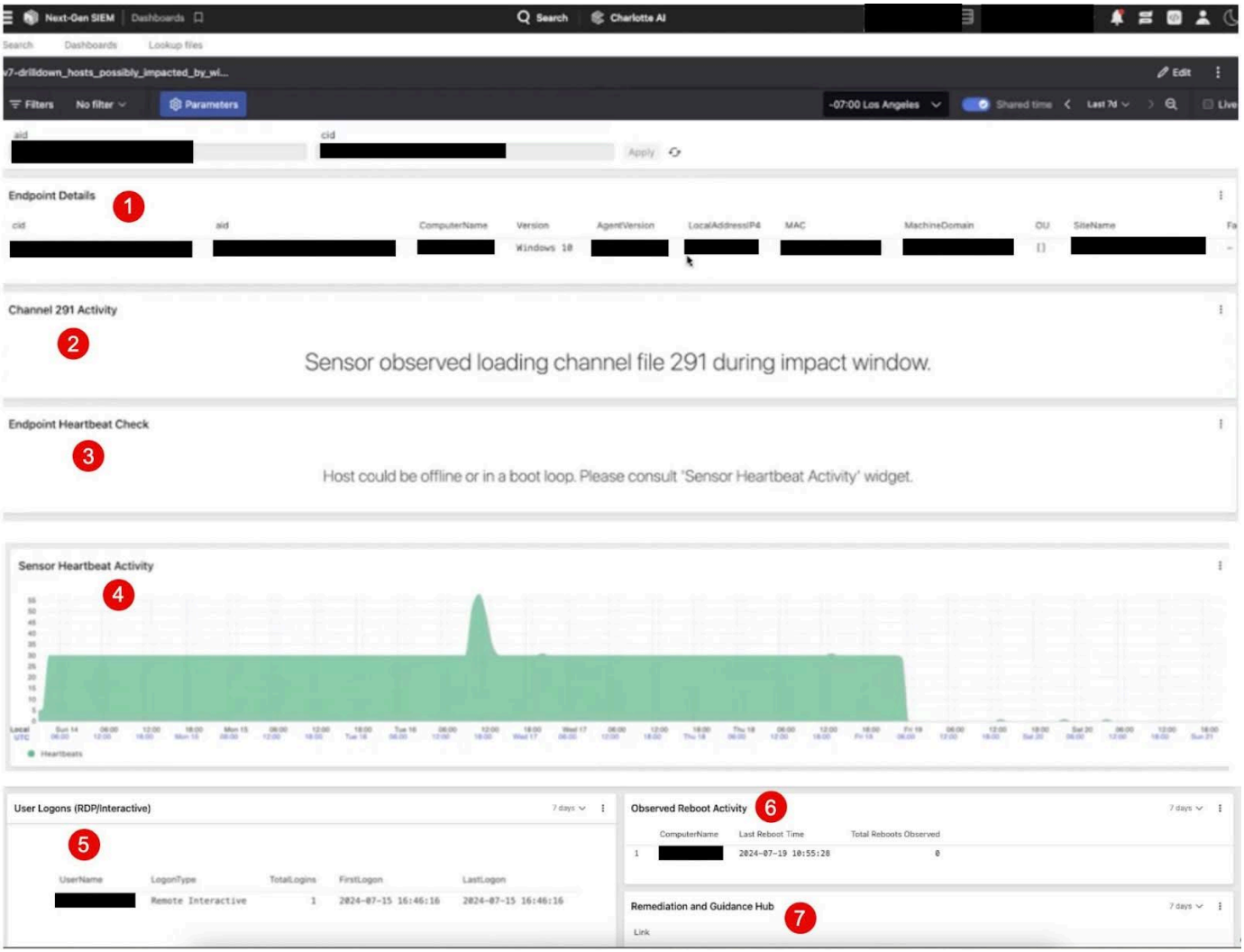
Drill-down dashboard

The **Drill-down dashboard** provides additional information regarding the status of a specific host and can be accessed by selecting **See host details** from the 3-dot menu on either the **Hosts in potential boot loop** or **Impacted sensors by aid subset** widget on the dashboard (labeled 1).

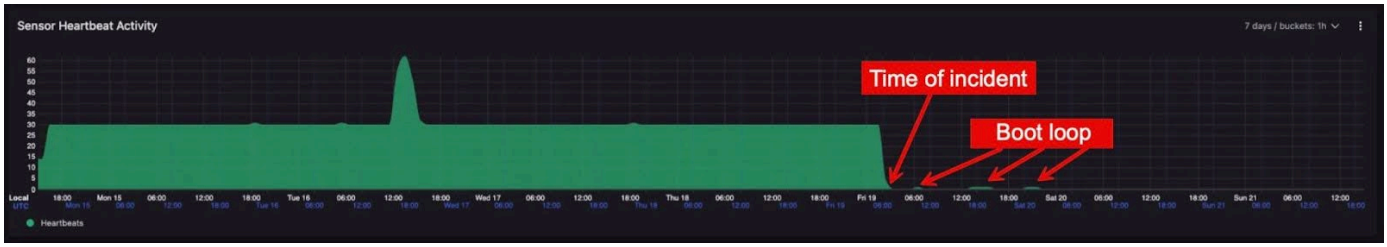


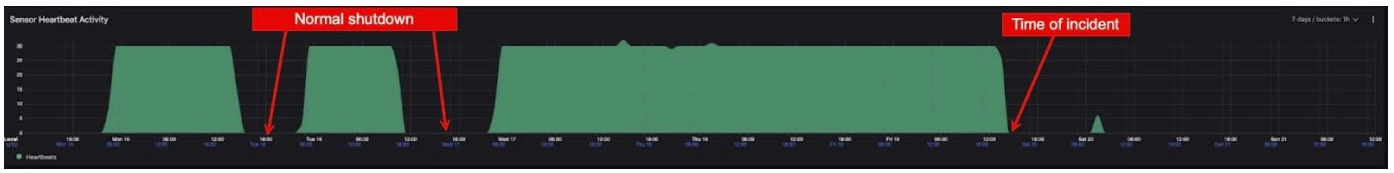
Drill-down dashboard widgets overview

- **Endpoint Details** (labeled 1): Shows details of the system selected from the drill-down dashboard.
- **Channel 291 Activity** (labeled 2): Shows whether the system was potentially impacted by the content update defect by displaying one of the below values:
    - *Sensor observed loading channel file 291 during impact window.*
    - *Sensor did not interact with channel file 291 during impact window.*
- **Endpoint Heartbeat Check** (labeled 3): Shows the status of the system's connection to the CrowdStrike cloud by displaying one of the below values:
    - *Host was seen online after impact window. Host is likely not impacted or has recovered.*
    - *Host could be offline or in a boot loop. Please consult 'Sensor Heartbeat Activity' widget.*
- **Sensor Heartbeat Activity** (labeled 4): Shows a timeline of how frequently the sensor has checked into the CrowdStrike cloud.
- **User Logons (RDP/Interactive)** (labeled 5): Shows the last observed interactive login on the system.
- **Observed Reboot Activity** (labeled 6): Shows the last observed reboot time and total number of reboots.
- **Remediation and Guidance Hub** (labeled 7): Provides links to remediation and guidance resources.



Examples of SensorHeartbeat activity in VERIFY status





Use case ideas

- Review the **Sensor Heartbeat Activity** last connection time to determine whether the sensor stopped reporting around the time of the content issue.

- Review **User Logons** to determine which users need support to remediate their systems.

Additional Information

- We are aware of some instances of false positives and negatives for sensors in the dashboard. We will continue to provide updated dashboards for improved accuracy.

Changelog

- 2024-07-23 01:15 UTC | Updated navigation to dashboard

Dashboard Logic Queries

Note: In order for these queries to function, you will need to update the **aidsubset** field to the RegEx wildcard .*, or you will encounter an error.
✅ .*
🚫 .

Query that powers Hosts in potential boot loop

```
// Version 8.6 - 4:06PM EST 7/21/24
#event_simpleName=/^(AgentOnline|SensorHeartbeat)$/ event_platform=Win
| cid=?cid
| @timestamp > 1721362140000
| ConfigIDBuild >= 18110
| regex(field=aid, regex=?aidsubset)
| groupBy([cid, aid], function=[
    selectLast([ComputerName]),
    { #event_simpleName=AgentOnline
      | [
         min(BootId, as=BootMin),
         max(BootId, as=BootMax),
         { selectFromMin(field=@timestamp, include=[@timestamp]) |
rename(@timestamp, as=Earliest) },
         { selectFromMax(field=@timestamp, include=[@timestamp]) |
rename(@timestamp, as=Latest) }
      ]},
    { #event_simpleName=SensorHeartbeat | [max(@timestamp,
as=LastHeartbeatTime)] }
```

```
  ], limit=max)
| RebootsSinceAgentOnline:= BootMax - BootMin
| RebootsSinceAgentOnline > 0
| RebootTimeRange := Latest - Earliest
| RebootsPerHour := RebootsSinceAgentOnline / RebootTimeRange * 3600000
| Uptime := LastHeartbeatTime - Latest
| case {
    (Uptime < 0) OR (Uptime!=*) | Uptime:=0;
    *
  }
// This is set to 10 mins but can be changed.
| Uptime < 600000
| formatDuration("Uptime", precision=1, as=LastOnlineDuration)
| round(field=RebootsPerHour, how=ceil, as=AvgRebootsPerHour)
| TimeSinceLastHeartbeat := now() - LastHeartbeatTime
| formatDuration("TimeSinceLastHeartbeat", precision=2)
| formatTime(field=LastHeartbeatTime, format="%c", as="LastHeartbeatTime")
| formatTime(field=Latest, format="%c", as="LastBootTime")
| match(file="aid_master_main.csv", field=aid, include=ProductType,
strict=false)
| $falcon/helper:enrich(field=ProductType)
| default(value="-", field=[ComputerName, ProductType, LastBootTime,
LastHeartbeatTime, AvgRebootsPerHour, LastOnlineDuration,
RebootsSinceAgentOnline, TimeSinceLastHeartbeat], replaceEmpty=true)
// Filter on values.
| wildcard(field=ComputerName, pattern=?ComputerName, ignoreCase=true)
| wildcard(field=aid, pattern=?aid, ignoreCase=true)
| ProductType=?ProductType
| groupBy([cid, aid, ComputerName, ProductType, LastHeartbeatTime,
LastBootTime, RebootsSinceAgentOnline, AvgRebootsPerHour,
LastOnlineDuration, TimeSinceLastHeartbeat], function=[], limit=max)
```

Query that powers Impacted sensors by aid subset

```
// Version 8.6 - 4:06PM EST 7/21/24

// Get ConfigStateUpdate and SensorHeartbeat events
#event_simpleName=/^(ConfigStateUpdate|SensorHeartbeat)$/
event_platform=Win
| cid=?cid

// Splitting aid list for large customers
| regex(field=aid, regex=?aidsubset)

// Parse ConfigStateUpdate and extract Channel File 291 version number;
accept all SensorHeartbeat events and rename timestamp field
| case{
  #event_simpleName=ConfigStateUpdate |
regex("\|1,123,(?<CFVersion>.*?)\|", field=ConfigStateData, strict=false)
| parseInt(CFVersion, radix=16);
  #event_simpleName=SensorHeartbeat | rename([[@timestamp, LastSeen]]);
}
// Create counters for events that occured within impact window on July
19, 2024
| case{
  #event_simpleName=ConfigStateUpdate | @timestamp>1721362140000 AND
@timestamp<1721370420000 | CSUcounter:=1;
  #event_simpleName=SensorHeartbeat | LastSeen>1721362140000 AND
LastSeen<1721370420000 | SHBcounter:=1;
  *;
}

// Aggregate results by Agent ID (aid) value and extract maximum Channel
File 291 version per Customer ID (cid)
| groupBy([cid], function=[groupby(aid, function=([
  {selectLast(CFVersion)},
  {selectFromMax(field="@timestamp", include=[@timestamp, ComputerName,
aip, LocalAddressIP4, ConfigBuild]) | rename(field="@timestamp",
```

```
as="LastSeen")},
  {#event_simpleName=SensorHeartbeat LastSeen>1721370420000 |
count(as=TotalSHB)},
  max(CSUcounter, as=CSUcounter),
  max(SHBcounter, as=SHBcounter)
]), limit=max),
max(CFVersion, as=MaxCFVersion)
], limit=max)

// Set default values for CSUcounter and SHBcounter
| default(value="0", field=[CSUcounter,SHBcounter])

// Parse sensor build number from ConfigBuild
| ConfigBuild=/\d+\.\d+\.(?<BuildNumber>\d+)\./

// Calculate time between LastSeen value of every Agent ID and current
time
| LastSeenDelta:=now()-LastSeen

// Calculate duration between LastSeen and now
| LastSeenDelta:=formatDuration("LastSeenDelta", precision=2)

// Enrich aggregation results with aid_master details if available
| aid=~match(file="aid_master_main.csv", column=[aid], strict=false)
| aid=~match(file="aid_master_details.csv", column=[aid],
include=[FalconGroupingTags, SensorGroupingTags, ChassisType],
strict=false)

// Move ProductType to human-readable format
| $falcon/helper:enrich(field=ProductType)
// Prepare ChassisType to be moved to human-readable format after Falcon
Helper update
| $falcon/helper:enrich(field=ChassisType)
```

```
// Set default values for systems not found in aid_master file
| default(value="-", field=[MachineDomain, OU, SiteName,
FalconGroupingTags, SensorGroupingTags, ChassisType, ProductType],
replaceEmpty=true)

// Set default value for CFVersion if there is no ConfigStateUpdate event
in window
| default(value="-", field=[CFVersion])

// Explain what hard coded timestamp values map to
/*
    1721362140000 is Friday, July 19, 2024 04:09:00 UTC and represents the
start of the impact window
    1721366820000 is Friday, July 19, 2024 05:27:00 UTC and represents the
end of the impact window
    1721370420000 is Friday, July 19, 2024 06:27:00 UTC and represents the
end of the impact window + 1 hour as a buffer
*/

// EVALUATIONS USED TO CHECK SENSOR CONDITION BASED ON ABOVE AGGREGATION
//
| case{
    // GOOD: Endpoint running sensor version below 7.11
    BuildNumber<18110 | Status:="OK" | Code:=8 | Details:="Endpoint
running version of Falcon sensor that is not impacted.";
    // GOOD: Accounts for systems beleived to be known good.
    test(CFVersion==MaxCFVersion) | Status:="OK" | Code:=1 |
Details:="Endpoint has latest channel file and is operational.";
    // GOOD: Accounts for systems that are belived to be offline during
impact window
    CSUcounter=0 AND SHBcounter=0 | Status:="OK"    | Code:=2 |
Details:="Endpoint was offline and did not receive channel file during
impact window.";
    // GOOD: Accounts for systems that are beleived to be online during
```

```
impact window, but did not interact with CF 291
    CSUcounter=0 AND SHBcounter=1 | Status:="OK"    | Code:=3 |
Details:="Endpoint was online and did not receive channel file during
impact window.";
    // HARD DOWN: Accounts for systems with CFVersion of 0, but NOT
checked in after impact window.
    CFVersion=0 AND LastSeen<1721370420000 | Status:="DOWN" | Code:=4 |
Details:="Endpoint has channel file version of 0 and has not checked-in
after impact window.";
    // POSSIBLE SELF-RECOVERY : Accounts for systems that interacted with
CF 291, but has checked in after impact window
    CSUcounter=1 AND LastSeen>1721370420000 AND TotalSHB>600 |
Status:="OK" | Code:=5 | Details:="Endpoint received channel file during
impact window and has checked-in after impact window a total reported
uptime of 20+ hours.";
    // POSSIBLE SELF-RECOVERY : Accounts for systems that interacted with
CF 291, but has checked in after impact window
    CSUcounter=1 AND LastSeen>1721370420000 AND TotalSHB>300 |
Status:="RECOVERY_VERY_LIKELY" | Code:=6 | Details:="Endpoint received
channel file during impact window and has checked-in after impact window a
total reported uptime of 10-20 hours.";
    // POSSIBLE SELF-RECOVERY : Accounts for systems that interacted with
CF 291, but has checked in after impact window
    CSUcounter=1 AND LastSeen>1721370420000 AND TotalSHB>150 |
Status:="RECOVERY_LIKELY" | Code:=7 | Details:="Endpoint received channel
file during impact window and has checked-in after impact window with a
total reported uptime of 5-10 hours.";
    // POSSIBLE SELF-RECOVERY OR BOOT LOOP: Accounts for systems that
interacted with CF 291, but has checked in after impact window
    CSUcounter=1 AND LastSeen>1721370420000 | Status:="VERIFY" | Code:=9 |
Details:="Endpoint received channel file during impact window and has
checked-in after impact window.";
    // HARD DOWN: Accounts for systems that interacted with CF 291 and
appear to be offline since time of impact
```

```
    CSUcounter=1 AND LastSeen<1721370420000 | Status := "DOWN" |  Code:=10
| Details:="Endpoint received channel file during impacted window, but
endpoint has NOT checked-in after impact window.";
    // UNKNOWN: Accounts for systems that status can not be determined
based on rules and telemetry
    *                               | Status:="UNKNOWN"    | Code:=0 |
Details:="Cannot determine endpoint status based on available telemetry.";
}

// Convert FirstSeen time to human-readable format; results in UTC
| FirstSeen:=formatTime(format="%F %T %Z", field="FirstSeen")

// Convert LastSeen time to human-readable format; results in UTC
| LastSeen:=formatTime(format="%F %T %Z", field="LastSeen")

// Create dynamic filters for use in dashboard
| Status=?Status
| wildcard(field=ComputerName, pattern=?ComputerName, ignoreCase=true)
| wildcard(field=aid, pattern=?aid, ignoreCase=true)
| ProductType=?ProductType

// Create one final groupBy for easier export to CSV
| groupby([aid, ComputerName, Status, Code, LastSeen, CFVersion,
MaxCFVersion, TotalSHB, LastSeenDelta, Details, AgentVersion, aip,
FalconGroupingTags, LocalAddressIP4, MAC, MachineDomain, OU, ProductType,
SensorGroupingTags, SiteName, SystemManufacturer,SystemProductName,
Version, ChassisType, FirstSeen, cid, event_platform], limit=max,
function=[])
```