CROWDSTRIKE

How to identify hosts possibly impacted by Windows crashes

Published Date: July 19, 2024

Objective

» Identify Microsoft Windows hosts potentially impacted by crashes

» Scope impact related to Tech Alert | Windows crashes related to Falcon Sensor | 2024-07-19

Applies To

» Supported versions of the Falcons sensor for Windows

» Supported versions of Microsoft Windows

» May be related to Tech Alert | Windows crashes related to Falcon Sensor | 2024-07-19

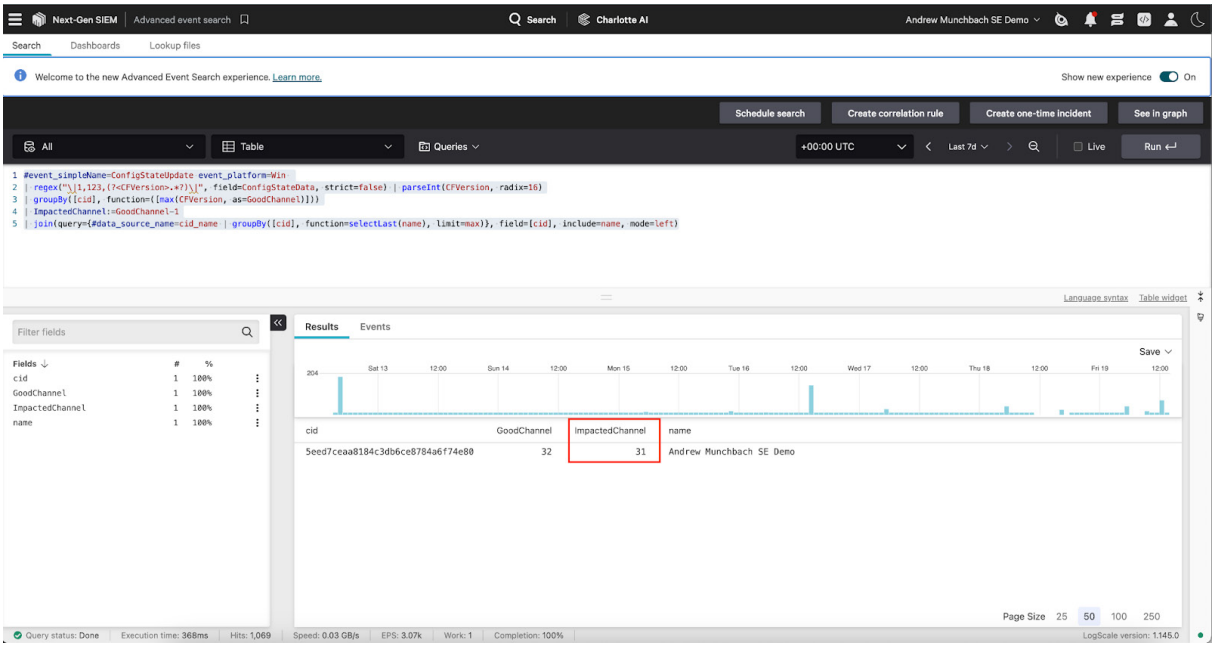
Procedure

Step 1: Determine Impacted Channel File

» Run the following query in Advanced Event Search with the search window set to seven days:

```
#event_simpleName=ConfigStateUpdate event_platform=Win
| regex("\|1,123,(?<CFVersion>.*?)\|", field=ConfigStateData, strict=false) |
parseInt(CFVersion, radix=16)
| groupBy([cid], function=([max(CFVersion, as=GoodChannel)]))
| ImpactedChannel:=GoodChannel-1
| join(query={#data_source_name=cid_name | groupBy([cid], function=selectLast(name),
limit=max)}, field=[cid], include=name, mode=left)
```

Please make note of the value listed in the column “ImpactedChannel.”



This number will differ slightly between Falcon tenants, but should be around 30.

Step 2: Execute query…

Execute the query below with the search window set to seven days. The query below will look for the following:

» Systems that were online during the impact window of 0400 - 0600 UTC 2024-07-19

» Systems that processed an update for Channel File 291 in the impact window of 0400 - 0600 UTC 2024-07-19

» Systems that last reported having loading the impacted channel file

» Systems that have not been seen in the past hour

You can add this query as a Scheduled Search (US-1 | US-2 | EU-1 | US-GOV-1) to run on a recurring interval of your choosing (every hour, for example).

IMPORTANT: Line 26 of this query needs to be edited with the value derived from the smaller query above. In our example instance, for this CID, we will use a value of 31. The line will read:

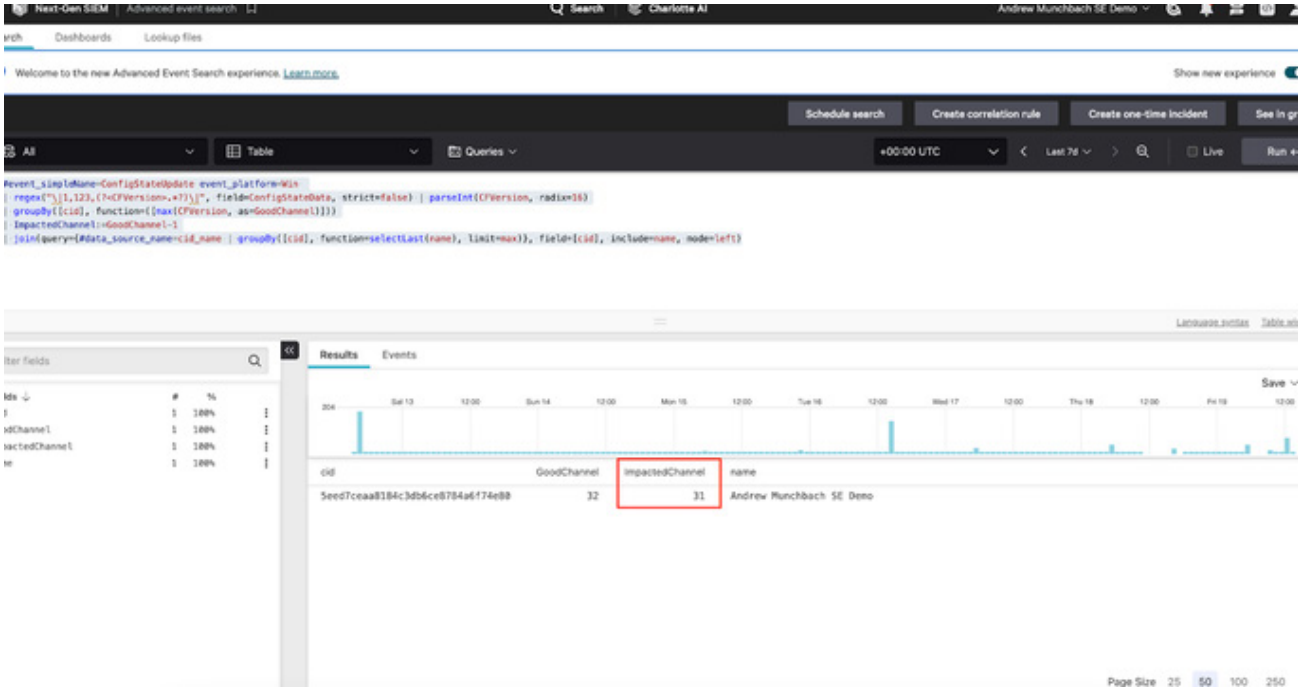
```
    [...]
    | in(field="CFVersion", values=[0,31])
    [...]
```

Please keep the number 0 in the “values” comma separated list.

```
// Get ConfigStateUpdate and SensorHeartbeat events
#event_simpleName=/^(ConfigStateUpdate|SensorHeartbeat)$/
event_platform=Win
| cid=?cid
// Narrow search to Channel File 291 and extract version number; accept
all SensorHeartbeat events
| case{
#event_simpleName=ConfigStateUpdate |
regex("\|1,123,(?<CFVersion>.*?)\|", field=ConfigStateData,
strict=false) | parseInt(CFVersion, radix=16);
#event_simpleName=SensorHeartbeat | rename([[@timestamp, LastSeen]]);
}
// Make sure both ConfigState update and SensorHeartbeat have happened
| selfJoinFilter(field=[cid, aid, ComputerName],
where=[{ConfigStateUpdate}, {SensorHeartbeat}])
// Aggregate results
| groupBy([cid], function=[groupby(aid, function=([
{selectFromMax(field="@timestamp", include=[CFVersion])},
{selectFromMax(field="@timestamp", include=[@timestamp]) |
rename(field="@timestamp", as="LastSeen")}
]), limit=max),
max(CFVersion, as=MaxCFVersion)
], limit=max)
// Perform check on selfJoinFilter
| CFVersion=* LastSeen=*
// Calculate time between last seen and now
| LastSeenDelta:=now()-LastSeen
// Only show the impacted channel
| in(field="CFVersion", values=[?Channel])
// Calculate duration between last seen and now
| LastSeenDelta:=formatDuration("LastSeenDelta", precision=2)
// Enrich aggregation with aid_master details
| aid=~match(file="aid_master_main.csv", column=[aid], strict=false)
| aid=~match(file="aid_master_details.csv", column=[aid],
include=[FalconGroupingTags, SensorGroupingTags], strict=false)
// Convert FirstSeen time to human-readable format
```

```
| FirstSeen:=formatTime(format="%F %T", field="FirstSeen")
// Move ProductType to human-readable format and add formatting
| $falcon/helper:enrich(field=ProductType)
| drop([Time])
| default(value="-", field=[MachineDomain, OU, SiteName,
FalconGroupingTags, SensorGroupingTags], replaceEmpty=true)
// Create conditions to check for impact
| case{
CFVersion=0 | Status:="VERIFY" | Details:="Endpoint channel file
version 0.";
test(CFVersion==(MaxCFVersion-1)) | Status := "CHECK" |
Details:="Endpoint has impacted channel file";
test(CFVersion==MaxCFVersion) | Status:="OK" | Details:="Endpoint has
latest channel file and is operational.";
test(CFVersion<(MaxCFVersion-1)) | Status:="OK" | Details:="Endpoint
has earlier channel file and is operational.";
* | Status:="UNKNOWN" |
Details:="Cannot determine status.";
}
// Convert FirstSeen time to human-readable format
| FirstSeen:=formatTime(format="%F %T", field="FirstSeen")
// Convert LastSeen time to human-readable format
| LastSeen:=formatTime(format="%F %T", field="LastSeen")
// Filter on status.
| Status=?Status
| wildcard(field=ComputerName, pattern=?ComputerName, ignoreCase=true)
// Create one final groupBy for easier export to CSV
| groupby([cid, aid, ComputerName, Status, FirstSeen, LastSeen,
CFVersion, MaxCFVersion, LastSeenDelta, Details, AgentVersion, aip,
event_platform, FalconGroupingTags, LocalAddressIP4, MAC, MachineDomain,
OU, ProductType, SensorGroupingTags, SiteName,
SystemManufacturer,SystemProductName, Version], limit=max, function=[])
```

The output of this query will show systems that have last reported running an impacted version of Channel File 291 that have not been seen in the past hour.



If the time window of one hour is too long, that can be adjusted in Line 26 of the query:

```
// Optional threshold; 3600000 is one hour

| LastSeenDelta>3600000
```

The value `3600000` is one hour in milliseconds. You can pick the threshold that best suits your needs.

Systems on this list should be evaluated to make sure they are not impacted.

Remediation instructions can be found in Tech Alert | Windows crashes related to Falcon Sensor | 2024-07-19.

CROWDSTRIKE

Formatted Code Blocks

Query 1

```
#event_simpleName=ConfigStateUpdate event_platform=Win

| regex("\|1,123,(?<CFVersion>.*?)\|", field=ConfigStateData, strict=false) |
parseInt(CFVersion, radix=16)

| groupBy([cid], function=([max(CFVersion, as=GoodChannel)]))

| ImpactedChannel:=GoodChannel-1

| join(query={#data_source_name=cid_name | groupBy([cid], function=selectLast(name),

limit=max)}, field=[cid], include=name, mode=left)
```

Query 2

```
// Get ConfigStateUpdate and SensorHeartbeat events

#event_simpleName=/^(ConfigStateUpdate|SensorHeartbeat)$/ event_platform=Win


| cid=?cid

// Narrow search to Channel File 291 and extract version number; accept all
SensorHeartbeat events

| case{

   #event_simpleName=ConfigStateUpdate | regex("\|1,123,(?<CFVersion>.*?)\|",
field=ConfigStateData, strict=false) | parseInt(CFVersion, radix=16);

   #event_simpleName=SensorHeartbeat | rename([[@timestamp, LastSeen]]);

}

// Make sure both ConfigState update and SensorHeartbeat have happened

| selfJoinFilter(field=[cid, aid, ComputerName], where=[{ConfigStateUpdate},
{SensorHeartbeat}])


// Aggregate results

| groupBy([cid], function=[groupby(aid, function=([

   {selectFromMax(field="@timestamp", include=[CFVersion])},

   {selectFromMax(field="@timestamp", include=[@timestamp]) | rename(field="@timestamp",
as="LastSeen")}

]), limit=max),

max(CFVersion, as=MaxCFVersion)

], limit=max)


// Perform check on selfJoinFilter

| CFVersion=* LastSeen=*
```

```
// Calculate time between last seen and now

| LastSeenDelta:=now()-LastSeen


// Only show the impacted channel

| in(field="CFVersion", values=[?Channel])


// Calculate duration between last seen and now

| LastSeenDelta:=formatDuration("LastSeenDelta", precision=2)


// Enrich aggregation with aid_master details

| aid=~match(file="aid_master_main.csv", column=[aid], strict=false)

| aid=~match(file="aid_master_details.csv", column=[aid],
include=[FalconGroupingTags, SensorGroupingTags], strict=false)


// Convert FirstSeen time to human-readable format

| FirstSeen:=formatTime(format="%F %T", field="FirstSeen")


// Move ProductType to human-readable format and add formatting

| $falcon/helper:enrich(field=ProductType)

| drop([Time])

| default(value="-", field=[MachineDomain, OU, SiteName, FalconGroupingTags,
SensorGroupingTags], replaceEmpty=true)


// Create conditions to check for impact

| case{

   CFVersion=0 | Status:="VERIFY" | Details:="Endpoint channel file version 0.";

   test(CFVersion==(MaxCFVersion-1)) | Status := "CHECK" | Details:="Endpoint has
impacted channel file";

   test(CFVersion==MaxCFVersion) | Status:="OK" | Details:="Endpoint has latest
channel file and is operational.";

   test(CFVersion<(MaxCFVersion-1)) | Status:="OK" | Details:="Endpoint has
earlier channel file and is operational.";

   *                              | Status:="UNKNOWN"    | Details:="Cannot
determine status.";

}


// Convert FirstSeen time to human-readable format

| FirstSeen:=formatTime(format="%F %T", field="FirstSeen")


// Convert LastSeen time to human-readable format

| LastSeen:=formatTime(format="%F %T", field="LastSeen")
```

```
// Filter on status.

| Status=?Status

| wildcard(field=ComputerName, pattern=?ComputerName, ignoreCase=true)

// Create one final groupBy for easier export to CSV

| groupby([cid, aid, ComputerName, Status, FirstSeen, LastSeen, CFVersion,
MaxCFVersion, LastSeenDelta, Details, AgentVersion, aip, event_platform,
FalconGroupingTags, LocalAddressIP4, MAC, MachineDomain, OU, ProductType,
SensorGroupingTags, SiteName, SystemManufacturer,SystemProductName, Version],
limit=max, function=[])
```