

Collect a custom metric for a custom device

Published: 2020-02-23

Custom metrics enable you to specify the metric data that the ExtraHop system collects from your network. After you create a custom metric with a trigger, you assign the trigger to your devices. This process is fairly straightforward for devices that are discovered by the ExtraHop system, but custom devices require additional processing.


When you have a custom metric that you want to collect for both discovered and custom devices, your trigger must include instructions on how to handle the different device types.

When a trigger is running on a discovered device, you can collect metrics through the `Flow.<role>.device` object. However, when a trigger is running on a custom device, you must collect metrics through the `Flow.<role>.customDevices` array. Both elements must be in any trigger that is assigned to custom and discovered devices.

In the following procedure, we show you how to modify an example trigger that collects a custom metric for HTTP 404 errors. The original trigger was written for only discovered devices, but we show you how to add parameters to collect the metric for custom devices as well.

Before you begin

- This topic assumes that you already have a custom metric that you want to assign to a custom device. For more information, see [Create a custom metric](#), and [Build a trigger to collect custom metrics for HTTP 404 errors](#), and [Create a custom device](#).
- You must have access to an ExtraHop Discover appliance with a user account that has limited write or full write privileges.
- You must have experience writing JavaScript code.

1. Log into the Web UI on the ExtraHop Discover or Command appliance.
2. Click the System Settings icon  and then click **Triggers**.
3. Click the name of the trigger that collects the custom metric.
4. Click the **Editor** tab.
5. Modify the trigger script to collect the custom metric for custom devices.

In this example, we will modify the following trigger code, which collects a custom metric for HTTP 404 errors on web servers that were discovered by the ExtraHop system:

```
if (HTTP.statusCode === 404){
  Flow.server.device.metricAddDetailCount(
    "404UriAndReferrer",
    "404:" + HTTP.uri + " | REFERRER:" + HTTP.referrer,
    1);
}
```

- a) First, add a statement that assigns the `Flow.server.device` object to a variable:

```
let server = Flow.server.device;
```

- b) Add a for loop that accesses each custom device that is acting as a server in the flow:

```
for (i = 0; i < Flow.server.customDevices.length; i++){
}
```

- c) Inside of the for loop, add an if statement that checks the `hasTrigger` property of each custom device to determine whether the trigger is running on the device:

```
if (Flow.server.customDevices[i]['hasTrigger']){
```

```
}

```

- d) Inside of the if statement, add a statement that reassigns the server variable to the device, and add another statement to break the for loop:

```
server = Flow.server.customDevices[i];
break;
```

- e) Finally, outside of the for loop, add a call to the `metricAddDetailCount` method to add the custom metric to the device:

```
server.metricAddDetailCount(
    "404UriAndReferrer",
    "404:" + HTTP.uri + " | REFERRER:" + HTTP.referrer,
    1);
```

Because the server variable is reassigned only if the trigger is running on a custom device, the code works for both discovered and custom devices. The complete code for this example is shown below:

```
if (HTTP.statusCode === 404){
    let server = Flow.server.device;
    for (i = 0; i < Flow.server.customDevices.length; i++){
        if (Flow.server.customDevices[i]['hasTrigger']){
            server = Flow.server.customDevices[i];
            break;
        }
    }
    server.metricAddDetailCount(
        "404UriAndReferrer",
        "404:" + HTTP.uri + " | REFERRER:" + HTTP.referrer,
        1);
}
```

6. Assign the trigger to the custom device.