Build a trigger

Published: 2024-04-02

Triggers provide expanded functionality of your ExtraHop system. With triggers, you can create custom metrics, generate and store records, or send data to a third-party system. Because you write the trigger script, you control the actions taken by the trigger upon specified system events.

To build a trigger, you must create a trigger configuration, write the trigger script, and then assign the trigger to one or more metric sources. The trigger will not run until all actions are completed.

Before you begin
Log in to the ExtraHop system with a user account that has the full write privileges required to create triggers.

If you are new to triggers, familiarize yourself with the trigger planning process, which will help you narrow the focus of your trigger, or determine whether you need a build a trigger at all. Then, run through the process of building a trigger by completing the Triggers Walkthrough.

Configure trigger settings

The first step to building a trigger is to provide a trigger name, determine whether debugging is enabled, and most importantly, identify which system events the trigger will run on.

1. Log in to the ExtraHop system through `https://<extrahop-hostname-or-IP-address>`.
2. Click the System Settings icon and then click **Triggers**.
3. Click **Create**.
4. Specify the following trigger configuration settings:

Name

A name for the trigger.

Author

The name of the user that wrote the trigger. Default triggers display ExtraHop.

Description

An optional description of the trigger.

Assignments

The devices or device groups the trigger is assigned to. A trigger does not run until it is assigned to a device, and the trigger gathers metric data only from the devices to which it is assigned.

Warning: Running triggers on unnecessary devices and networks exhausts system resources. Minimize performance impact by assigning a trigger only to the specific sources that you need to collect data from.

Important: Triggers with the following events run whenever the event occurs. Triggers that only run on these events cannot be assigned to devices or device groups.

- ALERT_RECORD_COMMIT
- DETECTION_UPDATE
- METRIC_CYCLE_BEGIN
- METRIC_CYCLE_END
- METRIC_RECORD_COMMIT
- NEW_APPLICATION
- NEW_DEVICE
- SESSION_EXPIRE

© 2024ExtraHop Networks, Inc. All rights reserved.

- TIMER_30SEC

Enable debug log

A checkbox that enables or disables debugging. If you add debug statements to the trigger script, this option enables you to view debug output in the debug log when the trigger is running.

Events

The events on which the trigger runs. The trigger runs whenever one of the specified events occurs on an assigned device; therefore, you must assign at least one event to your trigger. You can click in the field or begin typing an event name to display a filtered list of available events.

Advanced options

Advanced trigger options vary by the selected events. For example, if you select the `HTTP_RESPONSE` event, you can set the number of payload bytes to buffer on those events.

Write a trigger script

The trigger script specifies the instructions the trigger will carry out when a system event configured for the trigger occurs.

Before you begin

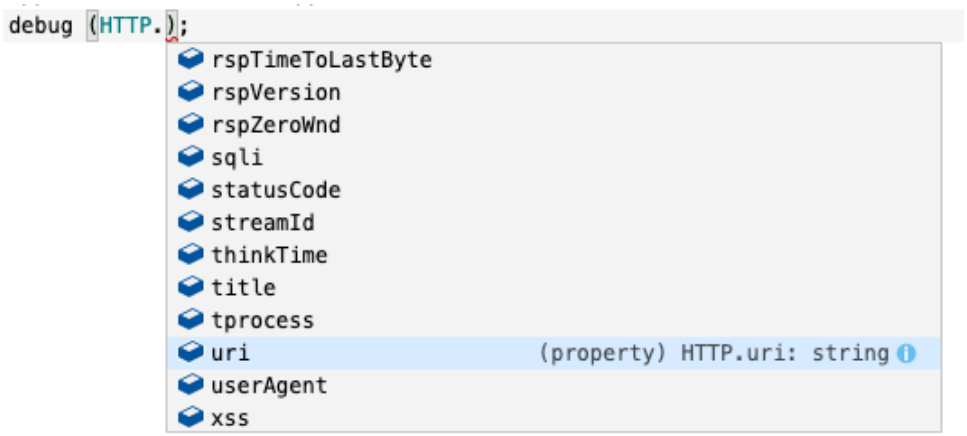
We recommend that you open the ExtraHop Trigger API Reference, which contains the events, methods, and properties you need for your trigger. A link is also available from the trigger editor window in the ExtraHop system.

1. Log in to the ExtraHop system through `https://<extrahop-hostname-or-IP-address>`.
2. Click the System Settings icon, and then click **Triggers**.
3. Click **Create**.
4. In the right pane, type the trigger script in JavaScript-like syntax with events, methods, and properties from the ExtraHop Trigger API Reference.

   The following figure shows a sample script entered on the Editor tab:

```
1  if (HTTP.uri.match("seattle")){
2      Application("Seattle App").commit();
3      debug (HTTP.uri);
4  }
```

The editor provides an autocomplete feature that displays a list of properties and methods based on the selected class object. For example, type a class name and then type a dot (.) to display a list of available properties and methods as shown in the following figure:

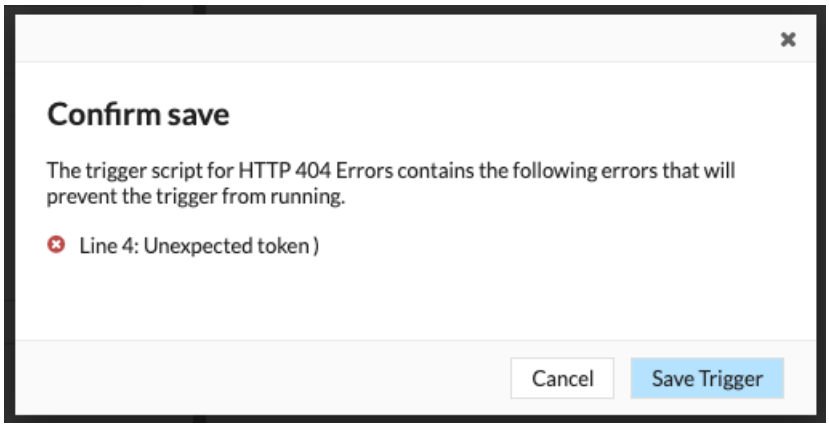


5. Click **Save**.

   The editor provides syntax validation of your script. When you save the trigger, the validator calls out any invalid actions, syntax errors, or deprecated elements in the script. If available, the validator displays replacements for deprecated elements.

   Warning: To avoid poor trigger performance, incorrect results, or a trigger that does not function, we strongly recommended that you fix the code or replace the deprecated element.

   The following figure shows a sample error message generated by the syntax validator:



Advanced trigger options

You must configure triggers to run on at least one event. Depending on the selected event, the Create Trigger pane displays advanced configuration options. For example, selecting the HTTP_RESPONSE event enables you to set the number of payload bytes to buffer each time that event occurs on the system.

The following table describes available advanced options and the events that support each option.

Option	Description	Supported events
Bytes Per Packet to Capture	Specifies the number of bytes to capture per packet. The capture starts with the first byte in the packet. Specify this option only if the trigger script performs packet capture. A value of 0 specifies that the capture should collect all bytes in each packet.	All events are supported except the following list: • `ALERT_RECORD_COMMIT` • `METRIC_CYCLE_BEGIN` • `METRIC_CYCLE_END` • `FLOW_REPORT` • `NEW_APPLICATION` • `NEW_DEVICE` • `SESSION_EXPIRE`
L7 Payload Bytes to Buffer	Specifies the maximum number of payload bytes to buffer. **Note:** If multiple triggers run on the same event, the trigge with the highest L7 Paylo Bytes to Buffer value determines the maximum payload for that event for each trigger.	• `CIFS_REQUEST` • `CIFS_RESPONSE` • `HTTP_REQUEST` • `HTTP_RESPONSE` • `ICA_TICK` • `LDAP_RESPONSE`
Clipboard Bytes	Specifies the number of bytes to buffer on a Citrix clipboard transfer.	• `ICA_TICK`
Metric cycle	Specifies the length of the metric cycle, expressed in seconds. The following values are valid: • `30sec` • `5min` • `1hr` • `24hr`	• `METRIC_CYCLE_BEGIN` • `METRIC_CYCLE_END` • `METRIC_RECORD_COMMIT`
Metric types	Specifies the metric type by the raw metric name, such as `extrahop.device.http_server`.	• `ALERT_RECORD_COMMIT` • `METRIC_RECORD_COMMIT`

Option	Description	Supported events
	Specify multiple metric types in a comma-delimited list.	
Run trigger on each flow turn	Enables packet capture on each flow turn. Per-turn analysis continuously analyzes communication between two endpoints to extract a single payload data point from the flow. If this option is enabled, any values specified for the **Client matching string** and **Server matching string** options are ignored.	• `SSL_PAYLOAD` • `TCP_PAYLOAD`
Client Port Range	Specifies the client port range. Valid values are 0 to 65535.	• `SSL_PAYLOAD` • `TCP_PAYLOAD` • `UDP_PAYLOAD`
Client Bytes to Buffer	Specifies the number of client bytes to buffer. The value of this option cannot be set to 0 if the value of the **Server bytes to buffer** option is also set to 0.	• `SSL_PAYLOAD` • `TCP_PAYLOAD`
Client Buffer Search String	Specifies the format string that indicates when to begin buffering client data. Returns the entire packet upon a string match. You can specify the string as text or hexidecimal numbers. For example, both `ExtraHop` and `\x45\x78\x74\x72\x61\x48\x6F\x70` are equivalent. Hexidecimal numbers are not case sensitive. Any value specified for this option is ignored if the **Per Turn** or **Run**	• `SSL_PAYLOAD` • `TCP_PAYLOAD` • `UDP_PAYLOAD`

Option	Description	Supported events
	trigger on all UDP packets option is enabled.	
Server Port Range	Specifies the server port range. Valid values are 0 to 65535.	• `SSL_PAYLOAD` • `TCP_PAYLOAD` • `UDP_PAYLOAD`
Server Bytes to Buffer	Specifies the number of server bytes to buffer. The value of this option cannot be set to 0 if the value of the **Client bytes to buffer** option is also set to 0.	• `SSL_PAYLOAD` • `TCP_PAYLOAD`
Server Buffer Search String	Specifies the format string that indicates when to begin buffering server data. You can specify the string as text or hexidecimal numbers. For example, both `ExtraHop` and `\x45\x78\x74\x72\x61\x48\x6F\x70` are equivalent. Hexidecimal numbers are not case sensitive. Any value specified for this option is ignored if the **Per Turn** or **Run trigger on all UDP** option is enabled.	• `SSL_PAYLOAD` • `TCP_PAYLOAD` • `UDP_PAYLOAD`
Run trigger on all UDP packets	Enables capture of all UDP datagrams.	• `UDP_PAYLOAD`
Run FLOW_CLASSIFY on expiring, unclassified flows	Enables running the event upon expiration to accumulate metrics for flows that were not classified before expiring.	• `FLOW_CLASSIFY`
External types	Specifies the types of external data the trigger processes. The trigger only runs if the payload contains a type field with one of the specified values. Specify multiple types in a comma-separated list.	1. `EXTERNAL_DATA`