Create a detection notification rule

Published: 2025-04-07

Create a notification rule if you want to receive a notification about detections that match specific criteria.

Video: See the related training: Configure Detection Notifications

When a detection that matches your criteria is generated, a notification is sent with information from the detection card.

You can configure the system to send an email to a recipient list or call a specific webhook. RevealX 360 users can create a notification rule that calls a webhook to export detection data to a configured integration.

Note: (RevealX 360 only) If you are creating a notification rule to export detection data to a SIEM integration, create the notification directly from the Integrations page in the Administration settings to pre-populate notification rule fields.

Before you begin

- Users must be granted NDR or NPM module access and have full write privileges or higher to complete the tasks in this guide.
- RevealX 360 requires a connection to ExtraHop Cloud Services to send notifications through email and webhooks. RevealX Enterprise requires a connection to ExtraHop Cloud Services to send notifications through email, but can send a notification through a webhook without a connection.
- Webhooks are sent over TCP 443 (HTTPS).
- Email notifications are sent through ExtraHop Cloud Services and might contain identifiable information such as IP addresses, usernames, hostnames, domain names, device names, or file names. RevealX Enterprise users who have regulatory requirements that prohibit external connections can configure notifications with webhook calls to send notifications without an external connection.
- Email notifications are sent from no-reply@notify.extrahop.com. Make sure to add this address to your list of allowed senders.

1. Log in to the ExtraHop system through `https://<extrahop-hostname-or-IP-address>`.
2. Click the System Settings icon and then click **Notification Rules**.
3. Click **Create**.
4. Click one of the following options:
   - For NDR modules, select **Security Detection**.
   - For NPM modules, select **Performance Detection**.
5. In the Name field, type a unique name for the notification rule.
6. In the Description field, add information about the notification rule.
7. In the Criteria section, click **Add Criteria** to specify criteria that will generate a notification.
   - **Recommended for Triage**
   - **Minimum Risk Score**
   - **Type**
   - **Category**
   - **MITRE Technique** (NDR only)
   - **Offender**
   - **Victim**
   - **Device Role**
   - **Participant**
   - **Site**

   The criteria options match the filtering options on the Detections page.

8. In the Target section, select how the notification will be sent from the following options:

Option	Description
Send Email	Send email notifications to a distribution list.
Custom Webhook	Send a JSON payload to a webhook URL.
Integration	Export detection data to a configured integration. For integrations, we recommend that ExtraHop administrators create detection notification rules from the Integrations page.

9. If you selected Send Email as the Target, complete the following steps:
   a) Specify individual email addresses, separated by a comma.
   b) Click **Save**.
10. If you selected Custom Webhook as the Target, complete the following steps:
    a) In the Payload URL field, type the URL of the webhook.
    b) Click **Show Advanced Connection Options** to configure the following:
       - In the Custom Headers section, click **Add Header** to specify custom key:value pairs.

         Custom headers are added to the header of the webhook HTTP POST request.
       - Select an authentication type.
         - No Authentication
         - Basic Authentication

           Enter the username and password for the target application.
         - Bearer Token

           Enter the access token for the target application.
       - Configure the connection method.
         - Direct Connection
           - Select to route the webhook through a configured global proxy. (RevealX Enterprise only.)
           - Select to skip server certificate verification.
         - Proxy through a connected sensor
           - Select the proxy sensor.
           - Select to skip server certificate verification.
           - Select to route the webhook through a global proxy that is configured for the selected sensor.

    c) Under Notification Behavior, select when the ExtraHop system will send notifications for a detection.
       - **Send for every detection update**

         Receive a notification every time the detection is updated.

         This selection is recommended if you are exporting detection data to a SIEM and want comprehensive visibility into detection activity.
       - **Send once per detection**

         Receive a single notification when a detection is created.

         This selection is optimal for notifying a group when a detection occurs without overwhelming the group with subsequent updates.

    d) Under Payload Options, select if you want to send the default payload or type in a custom JSON payload.

If you selected to send notifications once per detection under Notification Behavior, you must send a custom payload.

- **Default payload**

  Populate the webhook payload with a core set of detection fields.

  From the Add Payload Fields drop-down menu, you can click additional fields that you want to include in the payload.
- **Custom payload**

  Populate the webhook payload with custom JSON.

  You can edit the suggested custom payload in the **Edit Payload** window.

e) Click **Save**.

f) Click **Test Connection**.

A message titled Test Notification will be sent to the Payload URL to confirm the connection.

Note: After testing the connection, confirm that you received the notification in the target application. RevealX Enterprise displays an error message if the test notification was not successful.

11. In the Options section, the **Enable notification rule** checkbox is enabled by default. Deselect the checkbox to disable the notification rule.

When a detection matches the criteria, a notification is sent.

Webhook Notification Reference

This guide provides information about writing custom payloads for security or performance detection notifications with custom webhook or integration targets. The guide contains an overview of the Payload (JSON) interface, the default payload for webhook targets, a list of payload fields you can add to the default payload, and examples of JSON structure for common webhook targets, such as Slack, Microsoft Teams, and Google Chat.

Here are some considerations about webhook notifications:

- RevealX 360 cannot send webhook calls to endpoints on your internal network. Webhook targets must be open to external traffic.
- RevealX Enterprise must connect directly to webhook endpoints to send notifications.
- Webhook targets must have a certificate signed by a certificate authority (CA) from the Mozilla CA Certificate Program. See https://wiki.mozilla.org/CA/Included_Certificates for certificates from trusted public CAs.

For more information about notification rules, see Create a detection notification rule.

Payload JSON

ExtraHop webhooks are formatted in JSON, powered by the Jinja2 templating engine. When you create a security or performance detection notification rule and select a custom webhook or integration as the target, you will have the opportunity to select a default payload or write your own custom payload.

Default payload

The default payload option is available when you select to send a notification for every detection update as the notification behavior for the webhook. The default payload contains the following base set of information about a detection.

```
{
    "title": "{{ title }}",
    "type": "{{ type }}",
    "src": {
```

```
        "type": "{{ src.type }}",
        "hostname": "{{ src.hostname }}",
        "ipaddr": "{{ src.ipaddr }}",
        "role": "{{ src.role }}",
        "endpoint": "{{ src.endpoint }}",
        "username": "{{ src.username }}",
        "device": {
            "oid": {{ src.device.oid }},
            "name": "{{ src.device.name }}",
            "ipaddrs": {{ src.device.ipaddrs | safe }},
            "macaddr": "{{ src.device.macaddr }}"
        }
    },
    "dst": {
        "type": "{{ dst.type }}",
        "hostname": "{{ dst.hostname }}",
        "ipaddr": "{{ dst.ipaddr }}",
        "role": "{{ dst.role }}",
        "endpoint": "{{ dst.endpoint }}",
        "username": "{{ dst.username }}",
        "device": {
            "oid": {{ dst.device.oid }},
            "name": "{{ dst.device.name }}",
            "ipaddrs": {{ dst.device.ipaddrs | safe }},
            "macaddr": "{{ dst.device.macaddr }}"
        }
    },
    "additional_participants": {{ additional_participants | safe }},
    "properties": {{ properties | safe }},
    "description": "{{ description }}",
    "categories_ids": {{ categories_ids | safe }},
    "mitre_techniques": {{ mitre_techniques | safe }},
    "recommended": "{{ recommended }}",
    "recommended_factors": {{ recommended_factors | safe }},
    "url": "{{ url }}",
    "risk_score": {{ risk_score }},
    "time": {{ time }},
    "id": {{ detection_id or id }}
}
```

You can modify the default payload by selecting fields from the Add Payload Fields drop-down menu. To make custom changes, you can change your payload option to **Custom Payload**, then edit the suggested payload in the **Edit Payload** window.

Custom payload
Select the custom payload option to edit suggested JSON for a notification rule webhook.

If you select to send a notification for every detection update under Notification Behavior, the suggested custom payload contains the following JSON:

```
{
    "title": "{{ title }}",
    "type": "{{ type }}",
    "src": {
        "type": "{{ src.type }}",
        "hostname": {{ src.hostname | safe }},
        "ipaddr": "{{ src.ipaddr }}",
        "role": "{{ src.role }}",
        "endpoint": "{{ src.endpoint }}",
        "username": "{{ src.username }}",
        "device": {
            "oid": {{ src.device.oid }},
            "name": {{ src.device.name | safe }},
```

```
            "ipaddrs": {{ src.device.ipaddrs | safe }},
            "macaddr": {{ src.device.macaddr | safe }}
        }
    },
    "dst": {
        "type": "{{ dst.type }}",
        "hostname": {{ dst.hostname | safe }},
        "ipaddr": "{{ dst.ipaddr }}",
        "role": "{{ dst.role }}",
        "endpoint": "{{ dst.endpoint }}",
        "username": "{{ dst.username }}",
        "device": {
            "oid": {{ dst.device.oid }},
            "name": {{ dst.device.name | safe }},
            "ipaddrs": {{ dst.device.ipaddrs | safe }},
            "macaddr": {{ dst.device.macaddr | safe }}
        }
    },
    "additional_participants": {{ additional_participants | safe }},
    "properties": {{ properties | safe }},
    "description": "{{ description }}",
    "categories_ids": {{ categories_ids | safe }},
    "mitre_techniques": {{ mitre_techniques | safe }},
    "recommended": "{{ recommended }}",
    "recommended_factors": {{ recommended_factors | safe }},
    "url": "{{ url }}",
    "risk_score": {{ risk_score }},
    "time": {{ time }},
    "id": {{ detection_id or id }}
}
```

If you select to send one notification per detection update under Notification Behavior, the suggested custom payload contains the following JSON:

```
{
    "title": "{{ title }}",
    "type": "{{ type }}",
    "url": "{{ url }}",
    "description": "{{ description }}",
    "api": {{ api | safe }},
    "categories_string": "{{ categories_string }}",
    "categories_array": {{ categories_array | safe }},
    "victims": {{ victims | safe }},
    "offenders": {{ offenders | safe }},
    "description_format": "{{ description_format }}",
    "victim_primary": {{ victim_primary | safe }},
    "offender_primary": {{ offender_primary | safe }}
}
```

Tip: Before you take the time to type a lengthy custom payload, we recommend that you test your connection to the webhook URL. That way you can be sure any issues are not due to a connection error.

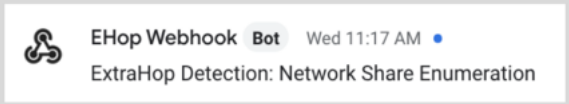
Syntax validation
The webhook editor provides JSON and Jinja2 syntax validation. If you type a line that includes incorrect JSON or Jinja2 syntax, an error appears under the Payload field with the error.

Variables
Detection variables are added to the payload by inserting the variable name between double sets of curly braces ({{ and }}).

For example, the sample in the payload includes a variable for the detection title:

```
"text": "ExtraHop Detection: {{title}}"
```

When a detection matches a notification rule with the variable, the variable is replaced by the detection title. For example, if the notification rule matches the detection for Network Share Enumeration, the variable is replaced with the title in the notification, similar to the following figure:



See a list of detection variables.

Filters
Filters enable you to modify a variable.

Passing JSON

If the variable returns a value that is formatted in JSON, the value is automatically escaped and translated into a string. If you want to pass valid JSON to your webhook target, you must specify the `safe` filter:

```
{{<variable> | safe }}
```

In the following example, the variable returns JSON-formatted detection data about participants directly to the webhook target:

```
{{api.participants | safe }}
```

IF statements
An IF statement can check whether a value is available for the variable. If the variable is empty, you can specify an alternative variable.

```
 {% if {{<variable>}} %}
```

In the following example, the IF statement checks if a value is available for the victim variable:

```
{% if victims %}
```

In the following example, the IF statement checks if an offender name is available. If there is no value for the offender name, the value for the offender IP address variable is returned instead.

```
{% if offender.name %}{{offender.name}}{%else%}{{offender.ipaddr}}
{% endif %}
```

FOR loops
A FOR loop can enable the notification to display an array of objects.

```
{% for <array-object-variable> in <array-variable> %}
```

In the following example, a list of offender names from the offenders array are displayed in the notification. An IF statement checks for more items in the array (`{% if not loop.last %}`) and adds a line break before printing the next value (`\n`). If an offender name is empty, the default filter returns “Unknown Name” for the value.

```
{% for offender in offenders %}
  {{offender.name | default ("Unknown Name")}}
  {% if not loop.last %}\n
  {% endif %}
```

```
{% endfor %}
```

Available detection variables
The following variables are available for webhook notifications about detections.

title: ***String***
The title of the detection.

description: ***String***
A description of the detection.

type: ***String***
The type of detection.

id: ***Number***
The unique identifier for the detection.

url: ***String***
The URL for the detection in the ExtraHop system.

risk_score: ***Number***
The risk score of the detection.

site: ***String***
The site where the detection occurred.

start_time_text: ***String***
The time that the detection started.

end_time_text: ***String***
The time that the detection ended.

categories_array: ***Array of Strings***
An array of categories that the detection belongs to.

categories_string: ***String***
A string that lists the categories that the detection belongs to.

properties: ***String***
A string that lists the properties linked to the detection.

recommended: ***Boolean***
The value is `true` if the detection has been recommended for triage.

recommended_factors: ***Array of Strings***
An array of factors that resulted in the detection being recommended for triage.

mitre_tactics: ***Array of Strings***
An array of MITRE tactic IDs associated with the detection.

mitre_tactics_string: ***String***
A string that lists the MITRE tactic IDs associated with the detection.

mitre_techniques: ***Array of Strings***
An array of MITRE technique IDs associated with the detection.

mitre_techniques_string: ***String***
A string that lists the MITRE technique IDs associated with the detection.

src:
The source participant in the detection. The source participant initiates the traffic associated with the detection.Each source participant object contains the following properties:

type:
The object type of the detection source. This value will be `device`, `ipaddr`, or `application`.

hostname:

The hostname linked to the detection source.

ipaddr:

The IP address linked to the detection source. This is the IP address that was seen during the detection.

role:

The detection role of the source. This value will be `offender` or `victim`.

endpoint:

The endpoint type of the source, according to the protocol. This value will be `client` or `sender`, depending on the protocol of the network traffic.

username:

The username linked to the detection source.

device:

The source device linked to the detection. This object is only present when the source type is device. Each object contains the following properties:

oid:

The unique ExtraHop object ID of the source device.

name:

The name of the source device.

ipaddrs:

The IP addresses associated with the source device. These IP addresses were not seen during the detection, but were previously associated with the device.

macaddr:

The MAC address of the source device.

dst:

The destination participant in the detection. The destination participant receives the initial traffic associated with the detection. Each destination participant object contains the following properties:

type:

The object type of the detection destination. This value will be `device`, `ipaddr`, or `application`.

hostname:

The hostname linked to the detection destination.

ipaddr:

The IP address linked to the detection destination. This is the IP address that was seen during the detection.

role:

The detection role of the destination. This value will be `offender` or `victim`.

endpoint:

The endpoint type of the destination, according to the protocol. This value will be `server` or `receiver`, depending on the protocol of the network traffic.

username:

The username linked to the detection destination.

device:

The destination device linked to the detection. This object is only present when the source type is device. Each object contains the following properties:

oid:

The unique ExtraHop object ID of the destination device.

name:
The name of the destination device.

ipaddrs:
The IP addresses associated with the destination device. These IP addresses were not seen during the detection, but were previously associated with the device.

macaddr:
The MAC address of the destination device.

offender_primary: ***Object***
(Deprecated) An object that identifies the primary offender and contains the following properties:

external: ***Boolean***
The value is `true` if the primary offender IP address is external to your network.

ipaddr: ***String***
The IP address of the primary offender.

name: ***String***
The name of the primary offender.

offenders: ***Array of Objects***
An array of offender objects associated with the detection. Each object contains the following properties:

external: ***Boolean***
The value is `true` if the offender IP address is external to your network.

ipaddr: ***String***
The IP address of the offender. Applies to detections with multiple offenders.

name: ***String***
The name of the offender. Applies to detections with multiple offenders.

victim_primary: ***Object***
(Deprecated) An object that identifies the primary victim and contains the following properties:

external: ***Boolean***
The value is `true` if the primary victim IP address is external to your network.

ipaddr: ***String***
The IP address of the primary victim.

name: ***String***
The name of the primary victim.

victims: ***Array of Objects***
An array of victim objects associated with the detection. Each object contains the following properties:

external: ***Boolean***
The value is `true` if the victim IP address is external to your network.

ipaddr: ***String***
The IP address of the victim. Applies to detections with multiple victims.

name: ***String***
The name of the victim. Applies to detections with multiple victims.

api: ***Object***
An object that contains all fields returned by the `GET /detections/{id}operation`. For more information, see the Introduction to the ExtraHop REST API.

Webhook Examples

The following sections provide JSON templates for common webhook targets.

Slack

After you create a Slack app and enable incoming webhooks for the app, you can create an incoming webhook. When you create an incoming webhook, Slack will generate the URL for you to enter in the Payload URL field in your notification rule.

The following example shows the JSON payload for a Slack webhook:

```
{
    "blocks": [
        {
            "type": "header",
            "text": {
                "type": "plain_text",
                "text": "Detection: {{ title }}"
             }
        },
        {
            "type": "section",
            "text": {
                "type": "mrkdwn",
                "text": "• *Risk Score:* {{ risk_score }}\n • *Category:*
 {{ categories_string }}\n • *Site:* {{ site }}\n • *Primary Offender:*
 {{ offender_primary.name}} ({{ offender_primary.ipaddr}})\n • *Primary
 Victim:* {{ victim_primary.name }} ({{ victim_primary.ipaddr }})\n"
            }
        },
        {
            "type": "section",
            "text": {
                "type": "plain_text",
                "text": "Detection ID: {{ id }}"
            },
            "text": {
                "type": "mrkdwn",
                "text": "<{{ url }}|View Detection Details>"
            }
        }
    ]
}
```

Microsoft Teams

You can add an incoming webhook to a Teams channel as a connector. After you configure an incoming webhook, Teams will generate the URL for you to enter in the Payload URL field in your notification rule.

The following example shows the JSON payload for a Microsoft Teams webhook:

```
{
   "type":"message",
   "attachments":[
      {
         "contentType":"application/vnd.microsoft.card.adaptive",
         "contentUrl":null,
         "content":{
            "$schema":"https://adaptivecards.io/schemas/adaptive-card.json",
            "type":"AdaptiveCard",
            "body":[
               {
                  "type":"ColumnSet",
                  "columns":[
                     {
                        "type": "Column",
```

```
                        "width":"16px",
                        "items":[
                           {
                              "type":"Image",
                              "horizontalAlignment":"center",
                              "url":"https://assets.extrahop.com/
favicon.ico",
                              "altText":"ExtraHop Logo"
                           }
                        ]
                     },
                     {
                        "type": "Column",
                        "width":"stretch",
                        "items":[
                           {
                              "type":"TextBlock",
                              "text":"ExtraHop RevealX",
                              "weight": "bolder"
                           }
                        ]
                     }
                  ]
               },
               {
                  "type":"TextBlock",
                  "text":"**{{ title }}**"
               },
               {
                  "type":"TextBlock",
                  "spacing":"small",
                  "isSubtle":true,
                  "wrap":true,
                  "text":"{{ description }}"
               },
               {
                  "type":"FactSet",
                  "facts":[
                     {
                        "title":"Risk Score:",
                        "value":"{{ risk_score }}"
                     },
                     {
                        "title":"Category:",
                        "value":"{{ categories_string }}"
                     },
                     {
                        "title":"Site:",
                        "value":"{{ site }}"
                     },
                     {
                        "title":"Primary Offender:",
                        "value":"{{ offender_primary.name }}
 ({{ offender_primary.ipaddr }})"
                     },
                     {
                        "title":"Primary Victim:",
                        "value":"{{ victim_primary.name }}
 ({{ victim_primary.ipaddr }})"
                     }
                  ]
               },
               {
                  "type":"ActionSet",
```

```
                  "actions":[
                     {
                        "type":"Action.OpenUrl",
                        "title":"View Detection Details",
                        "url":"{{ url }}"
                     }
                  ]
               }
            ]
         }
      }
   ]
}
```

Microsoft Teams with Adaptive Cards

You can configure a webhook to send Adaptive Cards, which enables you to customize the look of your incoming webhook in Teams.

However, it is important to note that we do not recommend the Adaptive Card for display in the Teams mobile app, because your content might not render as expected.

The following example shows the JSON payload for a Microsoft Teams webhook with an Adaptive Card:

```
{
   "type":"message",
   "attachments":[
      {
         "contentType":"application/vnd.microsoft.card.adaptive",
         "contentUrl":null,
         "content":{
            "$schema":"https://adaptivecards.io/schemas/adaptive-card.json",
            "type":"AdaptiveCard",
            "body":[
               {
                  "type":"ColumnSet",
                  "columns":[
                     {
                        "type": "Column",
                        "width":"16px",
                        "items":[
                           {
                              "type":"Image",
                              "horizontalAlignment":"center",
                              "url":"https://assets.extrahop.com/
favicon.ico",
                              "altText":"ExtraHop Logo"
                           }
                        ]
                     },
                     {
                        "type": "Column",
                        "width":"stretch",
                        "items":[
                           {
                              "type":"TextBlock",
                              "text":"ExtraHop Reveal(x)",
                              "weight": "bolder"
                           }
                        ]
                     }
                  ]
               },
```

```
                {
                   "type":"TextBlock",
                   "text":"**{{ title }}**"
                },
                {
                   "type":"TextBlock",
                   "spacing":"small",
                   "isSubtle":true,
                   "wrap":true,
                   "text":"{{ description }}"
                },
                {
                   "type":"FactSet",
                   "facts":[
                      {
                         "title":"Risk Score:",
                         "value":"{{ risk_score }}"
                      },
                      {
                         "title":"Category:",
                         "value":"{{ categories_string }}"
                      },
                      {
                         "title":"Site:",
                         "value":"{{ site }}"
                      },
                      {
                         "title":"Primary Offender:",
                         "value":"{{ offender_primary.name }}
 ({{ offender_primary.ipaddr }})"
                      },
                      {
                         "title":"Primary Victim:",
                         "value":"{{ victim_primary.name }}
 ({{ victim_primary.ipaddr }})"
                      }
                   ]
                },
                {
                   "type":"ActionSet",
                   "actions":[
                      {
                         "type":"Action.OpenUrl",
                         "title":"View Detection Details",
                         "url":"{{ url }}"
                      }
                   ]
                }
             ]
          }
       }
    ]
}
```

Google Chat

From a Google chat room, you can click the drop-down menu next to the room name and select Manage webhooks. After you add a webhook and name it, Google Chat will generate the URL for you to enter in the Payload URL field in your notification rule.

The following example shows the JSON payload for a Google Chat webhook:

```
{
```

```
    "cards": [
        {
            "header": {
                "title": "{{title}}"
            },
            "sections": [
                {
                    "widgets": [
                        {
                            "keyValue": {
                                "topLabel": "Risk score",
                                "content": "{{risk_score}}"
                            }
                        },
                        {
                            "keyValue": {
                                "topLabel": "Categories",
                                "content": "{{categories_string}}"
                            }
                        }
                        {% if offenders %}
                       ,{
                            "keyValue": {
                                "topLabel": "Offenders",
                                "contentMultiline": "true",
                                "content": "{% for offender in offenders %}
{% if offender.name %}{{offender.name}}{% else %}{{offender.ipaddr}}{% endif
 %}{% if not loop.last %}\n{% endif %}{% endfor %}"
                            }
                        }
                        {% endif %}
                        {% if victims %}
                       ,{
                            "keyValue": {
                                "topLabel": "Victims",
                                "contentMultiline": "true",
                                "content": "{% for victim in victims %}{%
 if victim.name %}{{victim.name}}{% else %}{{victim.ipaddr}}{% endif %}{% if
 not loop.last %}\n{% endif %}{% endfor %}"
                            }
                        }
                        {% endif %}
                    ]
                },
                {
                    "widgets": [
                        {
                            "buttons": [
                                {
                                    "textButton": {
                                        "text": "VIEW DETECTION DETAILS",
                                        "onClick": {
                                            "openLink": {
                                                "url": "{{url}}"
                                            }
                                        }
                                    }
                                }
                            ]
                        }
                    ]
                }
            ]
        }
```

```
    ]
}
```