Créer une règle de notification de détection

Publié: 2023-09-19

Créez une règle de notification si vous souhaitez recevoir une notification sur les détections qui correspondent à des critères spécifiques.

Lorsqu'une détection correspondant à vos critères est générée, une notification est envoyée avec les informations de la carte de détection.

Vous pouvez configurer le système pour qu'il envoie un courriel à une liste de destinataires ou qu'il appelle un webhook spécifique.

Avant de commencer

- Les utilisateurs doivent se voir accorder l'accès par le biais de la stratégie globale de contrôle d'accès aux détections et disposer de privilèges d' écriture complets ou supérieurs pour effectuer les tâches décrites dans ce guide.
- Reveal(x) 360 nécessite une connexion aux ExtraHop Cloud Services pour envoyer des notifications par courrier électronique et des webhooks. Reveal(x) Enterprise nécessite une connexion aux ExtraHop Cloud Services pour envoyer des notifications par courrier électronique, mais peut envoyer une notification par le biais d'un webhook sans connexion.
- Les notifications par courrier électronique sont envoyées via les services ExtraHop Cloud et peuvent contenir des informations identifiables telles que des adresses IP, des noms d'utilisateur, des noms d'hôte, des noms de domaine, des noms d'appareil ou des noms de fichier. Les utilisateurs de Reveal(x) Enterprise dont les exigences réglementaires interdisent les connexions externes peuvent configurer les notifications avec des appels webhook pour envoyer des notifications sans connexion externe.
- Reveal(x) 360 ne peut pas envoyer d'appels webhook à des points d'extrémité sur votre réseau interne. Les cibles webhook doivent être ouvertes au trafic externe.
- Les cibles des webhooks doivent disposer d'un certificat signé par une autorité de certification (CA) du Mozilla CA Certificate Program. Voir https://wiki.mozilla.org/CA/Included_Certificates pour les certificats d'autorités de certification publiques de confiance.
- Reveal(x) Enterprise doit se connecter directement aux points de terminaison des webhooks pour envoyer des notifications.
- Les notifications par courriel sont envoyées à partir de no-reply@notify.extrahop.com. Veillez à ajouter cette adresse à votre liste d'expéditeurs autorisés.

1. Connectez-vous au système ExtraHop via `https://<extrahop-hostname-or-IP-address>`.
2. Cliquez sur l'icône Paramètres système, puis sur **Règles de notification**.
3. Cliquez sur **Créer**.
4. Dans le champ Nom , saisissez un nom unique pour la règle de notification.
5. Dans le champ Description, ajoutez des informations sur la règle de notification.
6. Dans la section Type d'événement, sélectionnez **Détection**.
7. Dans la section Critères, cliquez sur **Ajouter un critère** pour spécifier les critères qui généreront une notification.
    - **Score de risque minimum**
    - **Type d'événement**
    - **Catégorie**
    - **Technique**
    - **Délinquant**
    - **Victime**
    - **Dispositif Rôle**
    - **Source d'information**
    - **Site**

Les options de critères correspondent aux options de filtrage de la page Détections.

8. Dans la section Actions, cliquez sur **Ajouter une action** pour spécifier le mode d'envoi de la notification.
   - Cliquez sur **Envoyer un courrier électronique** et indiquez des adresses électroniques individuelles, séparées par une virgule.
   - Cliquez sur **Appeler Webhook** et spécifiez les paramètres suivants :
     1. Dans le champ Payload URL, saisissez l'URL du webhook.
     2. Dans le champ Payload (JSON), saisissez la charge utile JSON qui sera envoyée à l'URL de la charge utile.

        Voir Référence pour les notifications par webhook pour des exemples de charge utile

        .
     3. (Facultatif) Dans la section En-têtes personnalisés, cliquez sur **Ajouter un en-tête** pour spécifier des paires clé/valeur personnalisées

        . Les en-têtes personnalisés sont ajoutés à l'en-tête de la requête HTTP POST du webhook

        .
     4. Cliquez sur **Save (Enregistrer)**.
     5. Cliquez sur **Test Connection**

        .Un message intitulé Test Notification sera envoyé à l'URL Payload pour confirmer la connexion.

        Note: Après avoir testé la connexion, confirmez que vous avez reçu la notification dans l'application cible.Reveal(x) Enterprise affiche un message d'erreur si la notification de test n'a pas réussi.
     6. Sélectionnez un type d'authentification.
        - **Pas d'authentification**
        - **Authentification de base**

          Saisissez le nom d'utilisateur et le mot de passe de l'application cible.
        - **Jeton de support**

          Entrez le jeton d'accès pour l'application cible.
9. Dans la section Options, cochez la case **Activer la règle de notification** pour activer la notification.

Lorsqu'une détection correspond aux critères, une notification est envoyée. Une seule détection ne génère jamais plus d'une notification par règle de notification.

Référence pour les notifications par webhook

Ce guide fournit des informations de référence pour vous aider à écrire la charge utile JSON pour les notifications basées sur les webhooks. Il contient une présentation de l'interface Payload (JSON), une liste des variables de détection disponibles pour les webhooks et des exemples de structure JSON pour des cibles webhooks courantes, telles que Slack, Microsoft Teams et Google Chat.

Pour plus d'informations sur les règles de notification, voir Créer une règle de notification de détection.

Charge utile JSON

Les webhooks ExtraHop sont formatés en JSON, grâce au moteur de template Jinja2. Lorsque vous créez une règle de notification et que vous sélectionnez l'option webhook, l'éditeur de webhook s'ouvre à droite et vous pouvez modifier la charge utile.

Vous pouvez modifier la charge utile par défaut avec des propriétés personnalisées ou copier un modèle JSON pour Slack, Microsoft Teams ou Google Chat, à partir de la section Exemples.

Par défaut, la charge utile contient un exemple de propriété `text`. L'exemple JSON de la figure ci-dessous envoie une notification avec le texte "ExtraHop Detection" suivi du titre de la détection qui remplace la variable.

Nous vous recommandons de tester votre connexion à l'URL du webhook avant de modifier la charge utile. Vous serez ainsi certain que les problèmes éventuels ne sont pas dus à une erreur de connexion.

Validation de la syntaxe

L'éditeur de webhook propose une validation syntaxique JSON et Jinja2. Si vous tapez une ligne contenant une syntaxe JSON ou Jinja2 incorrecte, une erreur apparaît sous le champ Payload avec l'erreur.

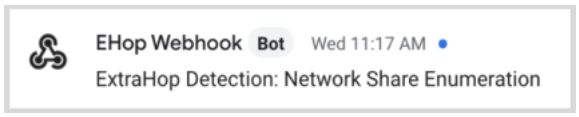
Variables

Les variables de détection sont ajoutées à la charge utile en insérant le nom de la variable entre deux séries d'accolades ({{ et }}).

Par exemple, l'échantillon figurant dans la charge utile comprend une variable pour le titre de la détection :

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

Lorsqu'une détection correspond à une règle de notification avec la variable, la variable est remplacée par le titre de la détection. Par exemple, si la règle de notification correspond à la détection de l'énumération des partages réseau, la variable est remplacée par le titre dans la notification, comme dans la figure suivante :



Voir la liste des variables de détection.

Les filtres

Les filtres permettent de modifier une variable.

Passer du JSON

Si la variable renvoie une valeur formatée en JSON, la valeur est automatiquement échappée et traduite en chaîne de caractères. Si vous souhaitez transmettre du JSON valide à votre cible de webhook, vous devez spécifier le filtre `safe`:

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

Dans l'exemple suivant, la variable renvoie des données de détection formatées JSON sur les participants directement à la cible du webhook :

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

Instructions IF

Une instruction IF permet de vérifier si une valeur est disponible pour la variable. Si la variable est vide, vous pouvez spécifier une autre variable.

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

Dans l'exemple suivant, l'instruction IF vérifie si une valeur est disponible pour la variable victime :

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

Dans l'exemple suivant, l'instruction IF vérifie si le nom de l'auteur de l'infraction est disponible. S'il n'y a pas de valeur pour le nom de l'auteur de l'infraction, la valeur de la variable adresse IP de l'auteur de l'infraction est renvoyée à la place.

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

Boucles FOR
Une boucle FOR peut permettre à la notification d'afficher un tableau d'objets.

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

Dans l'exemple suivant, la notification affiche une liste de noms de délinquants provenant du tableau "délinquants". Une instruction IF vérifie si le tableau contient d'autres éléments (`{% if not loop.last %}`) et ajoute un saut de ligne avant d'imprimer la valeur suivante (`\n`). Si le nom d'un délinquant est vide, le filtre par défaut renvoie la valeur "Nom inconnu".

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

Variables de détection disponibles
Les variables suivantes sont disponibles pour les notifications webhook concernant les détections.

title : *Chaîne*
Le titre de la détection.

detection : *Chaîne*
Description de la détection.

type : *Chaîne*
Le type de détection.

id : *Nombre*
L'identifiant unique de la détection.

url : *Chaîne*
L'URL de la détection dans le système ExtraHop.

risk_score : *Nombre*
Le score de risque de la détection.

site : *Chaîne*
Le site où la détection a eu lieu.

start_time_text : *Chaîne*
L'heure à laquelle la détection a commencé.

end_time_text : *Chaîne*
L'heure à laquelle la détection s'est terminée.

categories_array : *Tableau de chaînes*
Tableau des catégories auxquelles la détection appartient.

categories_string : *Chaîne*
Une chaîne qui énumère les catégories auxquelles la détection appartient.

mitre_tactics : *Tableau de chaînes*
Un tableau d'identifiants tactiques MITRE associés à la détection.

mitre_tactics_string : *Chaîne*

Une chaîne qui énumère les ID tactiques de MITRE associés à la détection.

mitre_techniques : *Tableau de chaînes*

Un tableau d'identifiants de techniques MITRE associés à la détection.

mitre_techniques_string : *Chaîne*

Une chaîne qui énumère les identificateurs de techniques de MITRE associés à la détection.

offender_primary : *Objet*

Un objet qui identifie l'auteur principal de l'infraction et qui contient les propriétés suivantes :

external : *Booléen*

La valeur est `true` si l'adresse IP de l'auteur principal de l'infraction est externe à votre réseau.

ipaddr : *Chaîne*

L'adresse IP de l'auteur principal de l'infraction.

name : *Chaîne*

Le nom du délinquant primaire.

offenders : *Tableau d'objets*

Un tableau d'objets délinquants associés à la détection. Chaque objet contient les propriétés suivantes :

external : *Booléen*

La valeur est `true` si l'adresse IP de l'auteur de l'infraction est extérieure à votre réseau.

ipaddr : *Chaîne*

L'adresse IP de l'auteur de l'infraction. S'applique aux détections avec plusieurs délinquants.

name : *Chaîne*

Le nom de l'auteur de l'infraction. S'applique aux détections comportant plusieurs délinquants.

victim_primary : *Objet*

Objet identifiant la victime principale et contenant les propriétés suivantes :

external : *Booléen*

La valeur est `true` si l'adresse IP de la victime principale est externe à votre réseau.

ipaddr : *Chaîne*

L'adresse IP de la victime principale.

name : *Chaîne*

Le nom de la victime principale.

victims : *Tableau d'objets*

Tableau d'objets victimes associés à la détection. Chaque objet contient les propriétés suivantes :

external : *Booléen*

La valeur est `true` si l'adresse IP de la victime est externe à votre réseau.

ipaddr : *Chaîne*

L'adresse IP de la victime. S'applique aux détections avec plusieurs victimes.

name : *Chaîne*

Le nom de la victime. S'applique aux détections comportant plusieurs victimes.

api : *Objet*

Un objet qui contient tous les champs renvoyés par `GET /detections/{id}operation`. Pour plus d'informations, voir Introduction à l'API REST d'ExtraHop.

Exemples de webhook

Les sections suivantes fournissent des modèles JSON pour des cibles webhook courantes.

Slack

Après avoir créé une application Slack et activé les webhooks entrants pour l'application, vous pouvez créer un webhook entrant. Lorsque vous créez un webhook entrant, Slack génère l'URL que vous devez saisir dans le champ Payload URL de votre règle de notification.

L'exemple suivant montre la charge utile JSON pour un webhook Slack :

```
{"blocks" : [ { "type" : "header", "text" : { "type" : "plain_text",
 "text" : "Détection : {{ titre }}" } }, { "type" : "section",
 "text" : {"type" : "mrkdwn", "text" : "- *Score de risque:*
 {{ risk_score }}\n - *Catégorie:* {{ categories_string }}\n - *Site:*
 {{ site }}\n - *Délinquant principal:* {{ délinquant_primaire.nom}}
 ({{ offender_primary.ipaddr}})\n - *Victime primaire:*
 {{ victim_primary.name }} ({{ victim_primary.ipaddr }})\n" } }, { "type" :
 "section", "text" : {"type" : "plain_text", "text" : "ID de détection :
 {{ id }}" }, "text" : {"type" : "mrkdwn", "text" : "<{{ url }}|Voir les
 détails de la détection>" } ] }
```

Microsoft Teams

Vous pouvez ajouter un webhook entrant à un canal Teams en tant que connecteur. Après avoir configuré un webhook entrant, Teams génère l'URL à saisir dans le champ Payload URL de votre règle de notification.

L'exemple suivant montre la charge utile JSON pour un webhook de Microsoft teams :

```
{ "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" : "Logo ExtraHop" } ] },
 { "type" : "Column", "width" : "stretch", "items" :[ {"type" :
 "TextBlock", "text" : "ExtraHop Reveal(x)", "weight" : "bolder" } ] },
 { "type" : "TextBlock", "text" : "**{{ titre }}**" }, { "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 }}
 ({{ victime_primaire.ipaddr }})" } ] }, { "type" : "ActionSet", "actions" :
[ { "type" : "Action.OpenUrl", "title" : "View Detection Details", "url" :
 "{{ url }}" } ] } ] } }
```

Chat Google

Dans un salon de discussion Google, vous pouvez cliquer sur le menu déroulant en regard du nom du salon et sélectionner Gérer les webhooks. Après avoir ajouté un webhook et l'avoir nommé, Google Chat génère l'URL que vous devez saisir dans le champ Payload URL de votre règle de notification.

L'exemple suivant montre la charge utile JSON pour un webhook Google Chat :

```
{"cards" : [ { "header" : {"title" : "{{title}}" }, "sections" :
 [ { "widgets" : [ { "keyValue" : {"topLabel" : "Score de risque",
 "content" : "{{risk_score}}" } }, { "keyValue" : {"topLabel" :
 "Catégories", "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" : "Victimes", "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}}" } } } } ] } ] } ] } ] } }
```