

Activer le suivi des détections


Publié: 2024-11-04

Le suivi des détections vous permet d'attribuer une détection à un utilisateur, de définir son statut et d'ajouter des notes. Vous pouvez suivre les détections directement dans le système ExtraHop, avec un système de billetterie externe tiers, ou avec les deux méthodes.



Note: Vous devez activer le suivi des tickets sur tous les capteurs connectés.

Avant de commencer

- Vous devez avoir accès à un système ExtraHop avec un compte utilisateur doté de **Privilèges d'administration** [↗](#).
 - Après avoir activé le suivi externe des tickets, vous devez **configurer le suivi des tickets par des tiers** en écrivant un déclencheur pour créer et mettre à jour des tickets sur votre système de billetterie, puis activez les mises à jour des tickets sur votre système ExtraHop via l'API REST.
 - Si vous désactivez le suivi externe des tickets, les informations de statut et de ticket des destinataires précédemment stockées sont converties en suivi de détection ExtraHop. Si le suivi de détection depuis le système ExtraHop est activé, vous pourrez consulter les tickets qui existaient déjà lorsque vous avez désactivé le suivi des tickets externes, mais les modifications apportées à ce ticket externe n'apparaîtront pas dans le système ExtraHop.
1. Connectez-vous aux paramètres d'administration du système ExtraHop via `https://<extrahop-hostname-or-IP-address>/admin`.
 2. Dans le Configuration du système section, cliquez sur **Suivi de la détection**.
 3. Sur la page de présentation, cliquez sur **Paramètres du système**  puis cliquez sur **Toute l'administration**.
 4. À partir du Paramètres de la console section, cliquez sur **Suivi de la détection**.
 5. Sélectionnez l'une des méthodes suivantes ou les deux pour suivre les détections :
 - Sélectionnez **Permettre aux utilisateurs d'ExtraHop de suivre les détections depuis le système ExtraHop**.
 - Sélectionnez **Activez des intégrations externes, telles que les systèmes SOAR ou de suivi des tickets, pour suivre les détections via l'API ExtraHop Rest**.
 6. Optionnel : Après avoir sélectionné l'option permettant d'activer les intégrations externes, spécifiez le modèle d'URL pour votre système de billetterie et ajoutez le `$ticket_id` variable à l'endroit approprié. Par exemple, saisissez une URL complète telle que `https://jira.example.com/browse/$ticket_id`. Le `$ticket_id` La variable est remplacée par l'identifiant du ticket associé à la détection. Une fois le modèle d'URL configuré, vous pouvez cliquer sur l'ID du ticket dans une détection pour ouvrir le ticket dans un nouvel onglet de navigateur.

Today 14:00
lasting an hour

83
RISK

LATERAL MOVEMENT

Status: **CLOSED**

Ticket ID: ✓ EX-4437

Assignee: hopuser

Suspicious CIFS Client File Share Access on AccountingLaptop

This device sent an excessive number of read requests over the Common Internet File System (CIFS) protocol. This anomaly indicates that the device might be compromised and is preparing files for data exfiltration.

Server linked to this anomaly:

- corpshare.example.com (192.168.6.179)

AccountingLaptop Activity Map

CIFS Metric	6-hour Snapshot	Peak Value	Expected Range	Deviation
Reads		1.13 K	0-1	112,500%

Prochaines étapes

Si vous avez activé les intégrations externes de suivi des tickets, vous devez passer à la tâche suivante :

- [Configurer le suivi des tickets par des tiers pour les détections](#)

Configurer le suivi des tickets par des tiers pour les détections

Le suivi des tickets vous permet de connecter les tickets, les alarmes ou les dossiers de votre système de suivi du travail aux détections ExtraHop. Tout système de billetterie tiers capable d'accepter les requêtes Open Data Stream (ODS), tel que Jira ou Salesforce, peut être lié aux détections ExtraHop.

Avant de commencer

- Tu dois avoir [sélectionné l'option de suivi de la détection par des tiers dans les paramètres d'administration](#).
- Vous devez avoir accès à un système ExtraHop avec un compte utilisateur doté de [Privilèges d'administration du système et des accès](#).
- Vous devez être familiarisé avec l'écriture de ExtraHop Triggers. Voir [éléments déclencheurs](#) et les procédures de [Créer un déclencheur](#).
- Vous devez créer une cible ODS pour votre serveur de suivi des tickets. Consultez les rubriques suivantes concernant la configuration des cibles ODS : [HTTP](#), [Kafka](#), [MongoDB](#), [syslog](#), ou [données brutes](#).
- Vous devez être familiarisé avec l'écriture de scripts d'API REST et disposer d'une clé d'API valide pour effectuer les procédures ci-dessous. Voir [Générer une clé API](#).



Rédigez un déclencheur pour créer et mettre à jour des tickets concernant les détections sur votre système de billetterie

Cet exemple montre comment créer un déclencheur qui exécute les actions suivantes :

- Créez un nouveau ticket dans le système de billetterie chaque fois qu'une nouvelle détection apparaît sur le système ExtraHop.
- Attribuer de nouveaux tickets à un utilisateur nommé `escalations_team` dans le système de billetterie.
- Exécuté chaque fois qu'une détection est mise à jour sur le système ExtraHop.
- Envoyez des mises à jour de détection via un flux de données ouvert (ODS) HTTP au système de billetterie.

L'exemple de script complet est disponible à la fin de cette rubrique.

1. Connectez-vous au système ExtraHop via `https://<extrahop-hostname-or-IP-address>`.

2. Cliquez sur l'icône des paramètres système  puis cliquez sur **DÉCLENCHEURS**.
3. Cliquez **Nouveau**.
4. Spécifiez un nom et une description facultative pour le déclencheur.
5. Dans la liste des événements, sélectionnez **MISE À JOUR DE DÉTECTION**.
L'événement DETECTION_UPDATE s'exécute chaque fois qu'une détection est créée ou mise à jour dans le système ExtraHop.
6. Dans le volet droit, spécifiez **Classe de détection**  paramètres d'un objet JavaScript. Ces paramètres déterminent les informations envoyées à votre système de billetterie.
L'exemple de code suivant ajoute l'identifiant de détection, la description, le titre, les catégories, les techniques et tactiques MITRE, ainsi que l'indice de risque à un objet JavaScript appelé `payload`:

```
const summary = "ExtraHop Detection: " + Detection.id + ": " +
  Detection.title;
const description = "ExtraHop has detected the following event on your
  network: " + Detection.description;
const payload = {
  "fields": {
    "summary": summary,
    "assignee": {
      "name": "escalations_team"
    },
    "reporter": {
      "name": "ExtraHop"
    },
    "priority": {
      "id": Detection.riskScore
    },
    "labels": Detection.categories,
    "mitreCategories": Detection.mitreCategories,
    "description": description
  }
};
```

7. Définissez ensuite les paramètres de requête HTTP dans un objet JavaScript situé sous l'objet JavaScript précédent.
L'exemple de code suivant définit une requête HTTP pour la charge utile décrite dans l'exemple précédent : définit une requête avec une charge utile JSON :

```
const req = {
  'path': '/rest/api/issue',
  'headers': {
    'Content-Type': 'application/json'
  },
  'payload': JSON.stringify(payload)
};
```

Pour plus d'informations sur les objets de requête ODS, voir [Classes de flux de données ouvertes](#) .

8. Enfin, spécifiez la requête HTTP POST qui envoie les informations à la cible ODS. L'exemple de code suivant envoie la requête HTTP décrite dans l'exemple précédent à une cible ODS nommée `ticket-server` :

```
Remote.HTTP('ticket-server').post(req);
```

Le code du déclencheur complet doit ressembler à l'exemple suivant :

```
const summary = "ExtraHop Detection: " + Detection.id + ": " +
  Detection.title;
```

```

const description = "ExtraHop has detected the following event on your
network: " + Detection.description
const payload = {
  "fields": {
    "summary": summary,
    "assignee": {
      "name": "escalations_team"
    },
    "reporter": {
      "name": "ExtraHop"
    },
    "priority": {
      "id": Detection.riskScore
    },
    "labels": Detection.categories,
    "mitreCategories": Detection.mitreCategories,
    "description": description
  }
};

const req = {
  'path': '/rest/api/issue',
  'headers': {
    'Content-Type': 'application/json'
  },
  'payload': JSON.stringify(payload)
};

Remote.HTTP('ticket-server').post(req);

```

Envoyer les informations de ticket aux détections via l'API REST

Après avoir configuré un déclencheur pour créer des tickets à détecter dans votre système de suivi des tickets, vous pouvez mettre à jour les informations relatives aux tickets sur votre système ExtraHop via l'API REST .

Les informations relatives aux tickets apparaissent dans les détections sur la page Détections du système ExtraHop. Pour plus d'informations, consultez [Détections](#) [☞](#) sujet.

L'exemple de script Python suivant extrait les informations de ticket d'un tableau Python et met à jour les détections associées sur le système ExtraHop.

```

#!/usr/bin/python3

import json
import requests
import csv


API_KEY = '123456789abcdefghijklmnop'
HOST = 'https://extrahop.example.com/'

# Method that updates detections on an ExtraHop system
def updateDetection(detection):
    url = HOST + 'api/v1/detections/' + detection['detection_id']
    del detection['detection_id']
    data = json.dumps(detection)
    headers = {'Content-Type': 'application/json',
              'Accept': 'application/json',
              'Authorization': 'ExtraHop apikey=%s' % API_KEY}
    r = requests.patch(url, data=data, headers=headers)
    print(r.status_code)
    print(r.text)

```

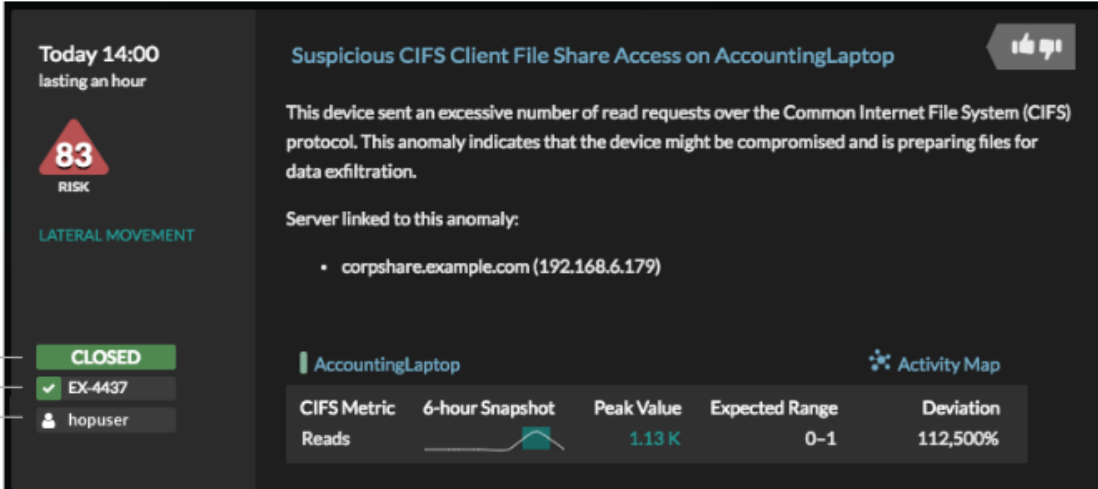
```
# Array of detection information
detections = [
    {
        "detection_id": "1",
        "ticket_id": "TK-16982",
        "status": "new",
        "assignee": "sally",
        "resolution": None,
    },
    {
        "detection_id": "2",
        "ticket_id": "TK-2078",
        "status": None,
        "assignee": "jim",
        "resolution": None,
    },
    {
        "detection_id": "3",
        "ticket_id": "TK-3452",
        "status": None,
        "assignee": "alex",
        "resolution": None,
    }
]

for detection in detections:
    updateDetection(detection)
```

 **Note:** Si le script renvoie un message d'erreur indiquant que la vérification du certificat TLS a échoué, assurez-vous que **un certificat fiable a été ajouté à votre sonde ou à votre console**. Vous pouvez également ajouter `verify=False` option permettant de contourner la vérification des certificats. Cependant, cette méthode n'est pas sûre et n'est pas recommandée. Le code suivant envoie une requête HTTP GET sans vérification du certificat :

```
requests.get(url, headers=headers, verify=False)
```

Une fois le suivi des tickets configuré, les détails des tickets sont affichés dans le volet gauche des détails de détection, comme dans la figure suivante :



Today 14:00
lasting an hour

83
RISK

LATERAL MOVEMENT


Suspicious CIFS Client File Share Access on AccountingLaptop

This device sent an excessive number of read requests over the Common Internet File System (CIFS) protocol. This anomaly indicates that the device might be compromised and is preparing files for data exfiltration.

Server linked to this anomaly:

- corpshare.example.com (192.168.6.179)

AccountingLaptop

CIFS Metric	6-hour Snapshot	Peak Value	Expected Range	Deviation
Reads		1.13 K	0-1	112,500%

Status: **CLOSED**

Ticket ID: **EX-4437**

Assignee: **hopuser**

État

État du ticket associé à la détection. Le suivi des tickets prend en charge les statuts suivants :

- Nouveau

- En cours
- Fermé
- Clôturé avec mesures prises
- Clôturé sans qu'aucune mesure n'ait été prise

ID du billet

L'identifiant du ticket associé à la détection dans votre système de suivi du travail. Si vous avez configuré un modèle d'URL, vous pouvez cliquer sur l'identifiant du ticket pour ouvrir le ticket dans votre système de suivi du travail.

Cessionnaire

Le nom d'utilisateur attribué au ticket associé à la détection. Les noms d'utilisateur en gris indiquent un compte qui n'est pas ExtraHop.