Créer une détection personnalisée

Publié: 2023-09-19

Les détections personnalisées vous permettent de spécifier les critères qui génèrent des détections sur le système ExtraHop. Les détections basées sur l'apprentissage automatique et les règles capturent les comportements inhabituels et les menaces courantes, mais en créant une détection personnalisée, vous pouvez vous concentrer sur les appareils et les comportements qui sont critiques pour votre réseau.

Lors de la création d'une détection personnalisée, vous devez créer un déclencheur qui identifie l'événement système et les conditions que le système doit surveiller, puis vous pouvez assigner le déclencheur aux appareils ou groupes d'appareils spécifiques que vous souhaitez surveiller. Lorsque l'événement se produit, une détection est générée.

Dans ce guide, nous fournissons les étapes et un exemple de script qui génère une détection personnalisée lorsque des connexions suspectes sont effectuées vers des sites web spécifiés via Windows PowerShell.

Avant de commencer

- Vous devez être familiarisé avec ExtraHop Déclencheurs. En particulier, tenez compte des meilleures pratiques suivantes lors de la rédaction de votre script et de l'attribution des déclencheurs.
- Vous devez disposer d'un compte utilisateur doté des privilèges requis pour créer des déclencheurs.
- Si vous disposez d'une console, créez un déclencheur sur la console et le déclencheur s'exécutera sur tous les capteurs connectés.

Créer un déclencheur pour générer des détections personnalisées

Les déclencheurs génèrent des détections personnalisées en appelant la fonction `commitDetection`dans le script du déclencheur.

Dans l'exemple suivant, le déclencheur génère une détection personnalisée lorsqu'un client PowerShell se connecte à un site web connu pour être un site de stockage de données exfiltrées.

Le déclencheur identifie les connexions PowerShell en recherchant les hachages JA3 du client SSL qui appartiennent à des clients PowerShell connus.

Si la connexion SSL est établie entre un client PowerShell et un hôte suspect, le déclencheur génère une détection. La détection inclut la version de PowerShell qui a initié la connexion, l'adresse IP du serveur et l'adresse IP du client.

Note: Pour plus d'informations sur la fonction `commitDetection`, consultez la Référence API du déclencheur.

1. Cliquez sur l'icône Paramètres système, puis sur **Déclencheurs**.
2. Cliquez sur **Créer**.
3. Spécifiez les paramètres de configuration des déclencheurs suivants :

Nom

Saisissez un nom pour votre déclencheur. Ce nom identifie votre déclencheur, et non la détection.

Dans notre exemple, nous allons entrer le nom : `Custom Detection : Connexion PowerShell à un site suspect`.

Description

(Facultatif) Saisissez la description du déclencheur. Cette description concerne le déclencheur et non la détection.

Dans notre exemple, nous allons saisir la description suivante : `Crée une détection chaque fois qu'un client PowerShell se connecte à pastebin,`

```
raw.githubusercontent.com ou githack. Les clients PowerShell sont
identifiés par des hachages JA3.
```

Événements

Sélectionnez l'événement sur lequel le déclencheur s'exécute.

Dans notre exemple, nous allons sélectionner l'événement SSL_OPEN. Cet événement se produit lorsqu'une connexion SSL est établie pour la première fois.

Affectations

Sélectionnez le dispositif ou le groupe de dispositifs que vous souhaitez surveiller. Dans un premier temps, affectez votre déclencheur à un seul dispositif à des fins de test. Une fois que vous avez confirmé que la détection personnalisée fonctionne correctement, affectez le déclencheur à un groupe de périphériques qui contient tous les périphériques que vous souhaitez surveiller

.Comme PowerShell est un outil de ligne de commande Windows, sélectionnez un serveur Microsoft pour tester le déclencheur. Une fois que vous avez confirmé que la détection personnalisée fonctionne correctement, modifiez l'affectation à un groupe de dispositifs qui contient tous vos serveurs Microsoft critiques. Pour plus d'informations sur la création de groupes de dispositifs, voir Créer un groupe de dispositifs.

4. Dans le volet de droite, saisissez le code qui détermine le moment où votre détection personnalisée est générée.

   Dans notre exemple, le code de déclenchement suivant identifie le moment où un client se connecte à pastebin, githubusercontent ou githack :

```
if(SSL.host.match(/pastebin/i) || SSL.host.match(/
raw.githubusercontent.com/i) || SSL.host.match(/githack/i)) {

}
```

5. Saisissez ensuite le code qui valide votre détection personnalisée. La fonction `commitDetection` doit être écrite dans le format suivant :

```
commitDetection('<detection type ID>', {
       title: '<title>',
       description: '<detection description>',
       categories: ['<category>'],
       riskScore: <risk score>,
       participants: [{
           object:<offender participant>,
           role: 'offender'
           }, {
           object: <victim participant>,
           role: 'victim'
           }],
       identityKey: '<identity key>',
       identityTtl: '<time period>',
       });
```

Entrez des valeurs pour chacun des paramètres suivants dans votre script.

Valeur	Description
ID du type de détection	Une chaîne unique qui identifie votre détection personnalisée. Cette chaîne ne peut contenir que des lettres, des chiffres et des traits de soulignement.
titre	Texte apparaissant en haut de la carte de détection.

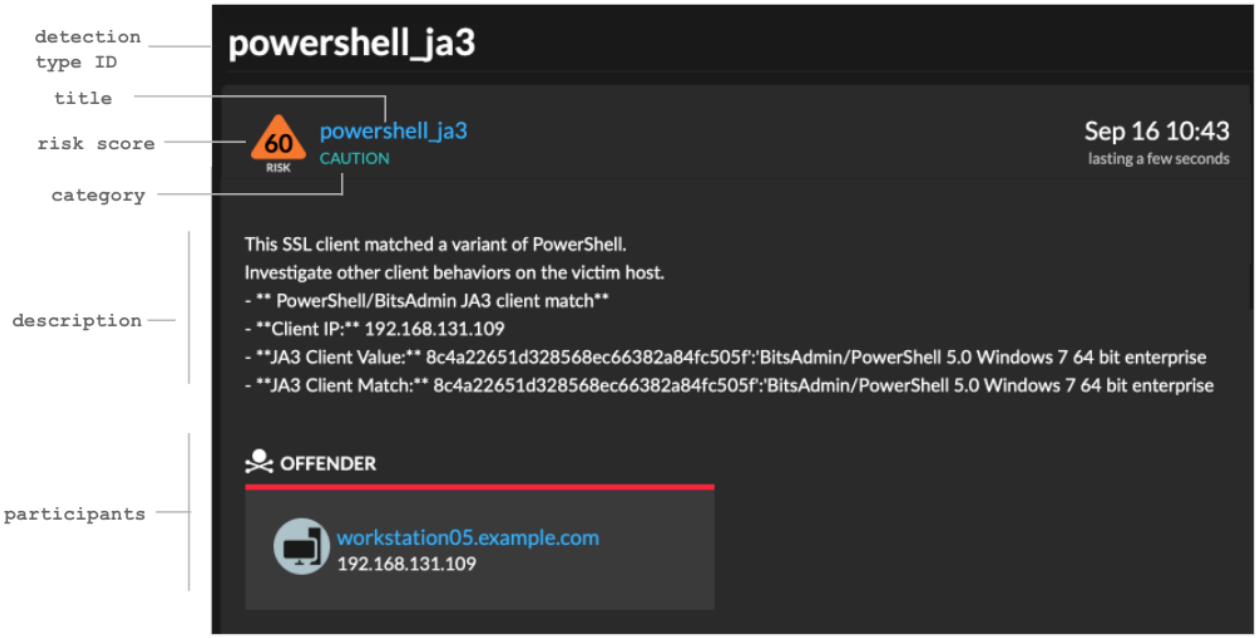
Valeur	Description
	Ce titre apparaît dans le catalogue des détections comme nom d'affichage de votre type de détection, précédé de `[custom]` .
description de la détection	Texte apparaissant sous le titre et la catégorie d'une fiche de détection. Tapez des informations sur l'événement qui génère la détection . Ce champ prend en charge le format markdown. Nous vous recommandons d'inclure des variables d'interpolation pour afficher des informations spécifiques sur votre détection. Par exemple, les variables `$(Flow.client.ipaddr)` et `$(Flow.server.ipaddr)` affichent l'adresse IP du client et du serveur dans le flux et `$(Flow.l7proto)` affiche le protocole L7. Incluez `\n` à la fin de chaque ligne de texte pour vous assurer que la description s'affiche correctement.
score de risque	Un nombre qui mesure la probabilité, la complexité et l'impact commercial d'une détection de sécurité. L'icône du score de risque apparaît en haut de la fiche de détection et est codée par couleur en fonction de la gravité : rouge (80-99), orange (31-79) ou jaune (1-30). Vous pouvez trier les détections en fonction du risque.
participant délinquant participant victime	Tableau d'objets identifiant les participants à la détection. Définissez le rôle du participant comme `'offender'` ou `'victim'` et fournissez une référence à un appareil, une adresse IP ou un objet d'application pour ce rôle. Par exemple, le tableau suivant identifie le serveur comme étant l'auteur de l'infraction et le client comme étant la victime dans un flux : `participants: [ { role: 'offender', object: Flow.server.device}, { role: 'victim', object: Flow.client.device } ]` Pour plus d'informations sur les dispositifs, les adresses IP et les objets d'application, voir la référence de l'API Trigger.
clé d'identité	Chaîne permettant d'identifier les détections en cours. Si plusieurs détections ayant la même clé d'identité et le même type de détection sont

Valeur	Description
	générées au cours de la période spécifiée par le paramètre `identityTtl`, les détections sont consolidées en une seule détection en cours .Créez une chaîne de clé d'identité unique en combinant les caractéristiques de la détection. Par exemple, la clé d'identité suivante est créée en combinant l'adresse IP du serveur et l'adresse IP du client : `identityKey: [Flow.server.ipaddr, Flow.client.ipaddr].join('!!')`
période de temps	Période de temps après la génération d'une détection pendant laquelle les détections en double sont regroupées dans une détection en cours. La période est réinitialisée et la détection ne prend fin qu'à l'expiration de la période . Les périodes suivantes sont valables : • `hour` • `day` • `week` La période par défaut est `hour`.

L'exemple suivant montre la section de script terminée.

```
commitDetection('powershell_ja3', {
        title:
'PowerShell / BitsAdmin Suspicious Connection',
        description:
"This SSL client matched a variant of PowerShell." + "\n"+
"Investigate other client behaviors on the victim host." + "\n"+
"- ** PowerShell/BitsAdmin JA3 client match**" + "\n"+
"- **Client IP:** " + Flow.client.ipaddr + "\n"+
"- **JA3 Client Value:** " + ja3 + "\n"+
"- **JA3 Client Match:** " + suspect_ja3_hashes[ja3],
        riskScore: 60,
        participants: [{
            object:Flow.client.device,
            role: 'offender'
            }],
        identityKey: [
            Flow.server.ipaddr,
            Flow.client.ipaddr,
            hash
        ].join('!!'),
        identityTtl: 'hour',
});
```

Ces valeurs apparaissent dans la carte de détection comme dans l'illustration suivante :



6. Cliquez sur **Enregistrer**, puis sur **Terminé**.

   Voir Exemple de déclencheur de détection personnalisé pour un script annoté complet.

Votre détection personnalisée sera ajoutée au catalogue de détection après la première exécution de votre déclencheur. Ajoutez des catégories de détection et des techniques MITRE à la détection à partir du catalogue de détection.

Création d'un type de détection personnalisé

Après avoir créé un déclencheur pour générer votre détection personnalisée, vous pouvez créer un type de détection personnalisé dans le catalogue des détections afin d'ajouter des informations supplémentaires à votre détection.

Vous pouvez spécifier un nom d'affichage et ajouter des catégories de détection pour vous aider à localiser votre détection sur la page Détections. Vous pouvez également ajouter des liens MITRE, qui permettent à votre détection personnalisée d'apparaître dans la matrice de la page Group by MITRE Technique.

1. Connectez-vous au système ExtraHop via `https://<extrahop-hostname-or-IP-address>`.
2. Cliquez sur l'icône System Settings (Paramètres système), puis sur **Detection Catalog (Catalogue de détection)**.
3. Sur la page Detection Catalog, effectuez l'une des étapes suivantes :
   - Si votre déclencheur a déjà été exécuté, le système ajoute automatiquement votre détection personnalisée au catalogue avec le nom d'affichage spécifié dans le déclencheur précédé de `[custom]`. Cliquez sur le type de détection à modifier.
   - Si votre type de détection n'a pas été créé, cliquez sur **Créer**.
4. Remplissez les champs suivants :

   Nom d'affichage

   Saisissez un nom unique pour le titre de la détection.

   ID du type de détection

   Saisissez la valeur que vous avez entrée pour l'ID du type de détection dans le déclencheur. Par exemple, si vous avez saisi : `commitDetection('network_segmentation_breach')`, l'ID

du type de détection est "network_segmentation_breach". Vous ne pouvez pas modifier l'ID du type de détection une fois que celui-ci est enregistré.

Auteur

Saisissez l'auteur de la détection personnalisée.

Technique MITRE

Dans la liste déroulante, sélectionnez une ou plusieurs techniques MITRE que vous souhaitez associer à la détection.

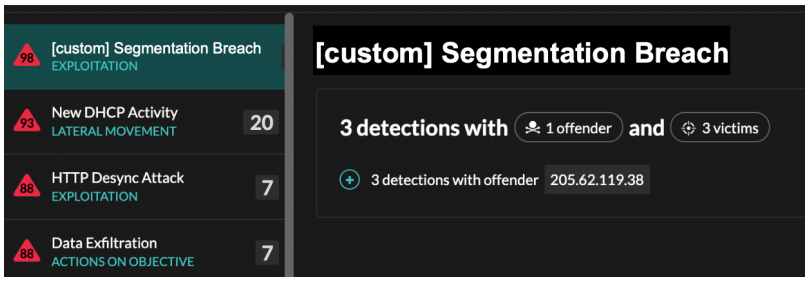
5. Cliquez sur **Save (Enregistrer)**.

Afficher les détections personnalisées

Vous pouvez afficher des détections personnalisées sur la page Détections avec d'autres détections intégrées.

Regrouper la page des détections par type. Toutes les détections de la liste des détections sont regroupées par type de détection.

Par exemple, si le nom d'affichage de votre détection est `[custom]Segmentation Breach`, l'entrée apparaîtra dans la liste des détections comme dans l'illustration suivante :



Dans le coin supérieur gauche de la page, sélectionnez MITRE **Map**. Les techniques MITRE liées à la détection personnalisée sont mises en évidence dans la matrice.

Prochaines étapes

Créer une règle de notification de détection. Vous pouvez, par exemple, configurer le système ExtraHop pour qu'il vous envoie un courrier électronique lorsque votre détection personnalisée se produit.

Exemple de déclencheur de détection personnalisé

Le script suivant est l'exemple PowerShell/JA3 complet auquel il est fait référence tout au long de ces instructions.

```
// Si le serveur est interne, quitter si ( ! Flow.server.ipaddr.isExternal )
 { return ; } // Si le nom d'hôte SSL n'est pas défini, quitter
 si(SSL.host === null) { return ; } // Continuer seulement si le nom
 d'hôte SSL appartient à l'un des sites suspects if(SSL.host.match(/
pastebin/i) || SSL.host.match(/raw.githubusercontent.com/i) ||
 SSL.host.match(/githack/i)) { // Liste des hachages PowerShell JA3
 courants let suspect_ja3_hashes = cache('suspect_ja3_hashes', () =>
 ({ '13cc575f247730d3eeb8ff01e76b245f' : 'PowerShell/BitsAdmin/PowerShell
 4.0 Windows Server 2012RT', '5e12c14bda47ac941fc4e8e80d0e536f' :
 'PowerShell/BitsAdmin/PowerShell 4.0 Windows Server 2012RT',
 '2c14bfb3f8a2067fbc88d8345e9f97f3':'PowerShell/BitsAdmin Windows Server
 2012RT', '613e01474d42ebe48ef52dff6a20f079':'PowerShell/BitsAdmin
```

```
 Windows Server 2012RT', '05af1f5ca1b87cc9cc9b25185115607d':'BitsAdmin/
PowerShell 5.0 Windows 7 64 bit enterprise',
 '8c4a22651d328568ec66382a84fc505f':'BitsAdmin/PowerShell 5.0 Windows
 7 64 bit enterprise', '235a856727c14dba889ddee0a38dd2f2':'BitsAdmin/
PowerShell 5.1 Server 2016', '17b69de9188f4c205a00fe5ae9c1151f':'BitsAdmin/
PowerShell 5.1 Server 2016', 'd0ec4b50a944b182fc10ff51f883ccf7':'PowerShell/
BitsAdmin (Microsoft BITS/7.8) Server 2016',
 '294b2f1dc22c6e6c3231d2fe311d504b':'PowerShell/
BitsAdmin (Microsoft BITS/7.8) Server 2016',
 '54328bd36c14bd82ddaa0c04b25ed9ad':'BitsAdmin/PowerShell 5.1 Windows 10',
 'fc54e0d16d9764783542f0146a98b300':'BitsAdmin/PowerShell 5.1 Windows 10',
 '2863b3a96f1b530bc4f5e52f66c79285':'BitsAdmin/PowerShell 6.0 Windows Server
 2012RT', '40177d2da2d0f3a9014e7c83bdeee15a':'BitsAdmin/PowerShell 6.0
 Windows Server 2012RT', '36f7277af969a6947a61ae0b815907a1':'PowerShell/
BitsAdmin Windows 7 32 bit enterprise', })) ; // Stocker le hachage JA3
 du client dans une variable const hash = SSL.ja3Hash ; // Interroger
 chaque hachage JA3 PowerShell for ( let ja3 in suspect_ja3_hashes )
 { // Si le hachage JA3 du client provient de PowerShell, // valider la
 détection if ( hash.includes(ja3) ) { commitDetection('PowerShell_JA3',
 { categories : ['sec.caution'], title : "PowerShell / BitsAdmin Suspicious
 Connection", // Spécifiez l'auteur de l'infraction comme étant l'objet
 de l'appareil du client participants : [ { role : 'offender', object :
 Flow.client.device } ], description :
                    "Ce client SSL correspond à une variante de PowerShell".
 + "\n "+ "Examiner les autres comportements du client sur l'hôte de la
 victime." + "\n-"+"- ** Correspondance client PowerShell/BitsAdmin JA3**"
 + "\n "+ "- **Client IP:** " + Flow.client.ipaddr + "\n "+ "- **Server
 IP:** " + Flow.server.ipaddr + "\n "+ "- **JA3 Client Value:** " + ja3
 + "\n "+ "- **JA3 Client Match:** " + suspect_ja3_hashes[ja3], // Créer
 la clé d'identité en combinant l'adresse IP du serveur, l'adresse IP du
 client et le hachage PowerShell JA3 identityKey : [ Flow.server.ipaddr,
 Flow.client.ipaddr, hash ].join('!!'), riskScore : 60, identityTtl :
 'hour' }) ; } } }
```