Hop supplémentaire
Guide des meilleures pratiques relatives aux déclencheurs

Publié: 2025-03-28

Rédaction d'un déclencheur collecter des métriques est un moyen puissant de surveiller les performances de votre application et de votre réseau. Cependant, les déclencheurs consomment des ressources système et peuvent affecter les performances du système. Un déclencheur mal écrit peut entraîner une charge système inutile. Avant de créer un déclencheur, vous devez évaluer ce que vous voulez qu'il accomplisse, identifier les événements et les appareils nécessaires pour extraire les données dont vous avez besoin et déterminer s'il existe déjà une solution.

Identifiez ce que vous voulez savoir

Identifiez clairement les informations que vous aimeriez connaître ou ce que vous souhaitez que le déclencheur accomplisse, par exemple dans les exemples suivants :

- Quand expireront mes certificats TLS ?
- Mon réseau est-il connecté à des ports non autorisés ?
- Combien de transactions sont lentes sur mon réseau ?
- Quelles données dois-je envoyer à Splunk via un flux de données ouvert ?

En identifiant le problème que vous essayez de résoudre, vous pouvez mieux cibler un déclencheur pour ne recueillir que les informations dont vous avez besoin et éviter de nuire inutilement aux performances.

Concentrez le déclencheur sur une fonction logique

Un déclencheur doit exécuter une fonction logique. Si vous avez besoin d'un déclencheur qui exécute une tâche différente, créez-en un deuxième. Un déclencheur qui exécute plusieurs tâches indépendantes consomme davantage de ressources.

Rechercher des métriques dans le catalogue de métriques

Assurez-vous de consulter les indicateurs intégrés dans le Catalogue métrique. Les métriques intégrées ne créent pas de charge supplémentaire sur le système et contiennent peut-être déjà les informations dont vous avez besoin.

Identifiez les événements du système qui génèrent les données que vous souhaitez collecter

Par exemple, un déclencheur qui surveille l'activité des applications cloud dans votre environnement peut s'exécuter sur les réponses HTTP et lors de l'ouverture et de la fermeture de connexions TLS.

Identifiez les appareils ou les réseaux que vous souhaitez surveiller et à partir desquels vous souhaitez collecter des statistiques

Un déclencheur consomme moins de ressources système si vous ciblez des appareils spécifiques plutôt que tous les appareils d'un type ou d'un groupe particulier. Par exemple, un déclencheur qui recherche les réponses lentes de votre catalogue en ligne doit être attribué uniquement aux serveurs HTTP qui gèrent les transactions du catalogue et non à tous les serveurs HTTP.

Déterminez comment vous souhaitez visualiser ou stocker les données collectées par le déclencheur

Par exemple, vous pouvez consulter les statistiques sur un tableau de bord, envoyer des enregistrements vers un espace de stockage des enregistrements ou envoyer des données vers un système tiers.

Déterminez si un déclencheur existe déjà

Il est possible qu'un déclencheur répondant à vos besoins ou pouvant être facilement modifié existe déjà ; commencez toujours par un déclencheur préexistant dans la mesure du possible. Recherchez dans Forums de la communauté ExtraHop pour les déclencheurs disponibles.

Optimisation de la configuration du déclencheur

Les options de configuration disponibles lorsque vous créez ou modifiez un déclencheur peuvent affecter les performances et l'efficacité du déclencheur. ExtraHop recommande les pratiques suivantes pour optimiser et améliorer les performances du déclencheur.

Profitez des options de déclencheur avancées

Si options de déclencheur avancées sont disponibles pour l'événement sur lequel le déclencheur s'exécute, nous vous recommandons de configurer les options applicables pour affiner le ciblage du déclencheur et améliorer les performances et les résultats. Par exemple, si vous créez un déclencheur qui s'exécute sur le `SSL_PAYLOAD` événement, vous pouvez spécifier des numéros de port minimum et maximum afin que le déclencheur collecte uniquement les données des transactions comprises dans la plage de ports spécifiée.

Attribuez le déclencheur à un minimum de ressources

Empêchez le déclencheur de s'exécuter inutilement en l'affectant au moins de sources possible, telles que des appareils. Ne pas sélectionner **Attribuer à tous les appareils** lors de l'attribution du déclencheur, et évitez les assignations à de grands groupes d'équipements.

Déboguer uniquement lors des tests

Le débogage ne doit être activé que lorsque vous travaillez activement sur votre déclencheur. Le débogage est utile pour vérifier que le déclencheur s'exécute et collecte les données attendues ; toutefois, le débogage consomme inutilement des ressources et doit être évité dans un environnement de production.

Écrire un script de déclencheur optimisé

Les sections suivantes présentent les meilleures pratiques générales en matière de codage ainsi que les directives relatives à l' utilisation de composants et d'objets d'API, tels que les tables de session et les enregistrements.

Appliquer les meilleures pratiques de codage

Lors de l'écriture d'un script de déclenchement, nous recommandons les meilleures pratiques de codage suivantes afin d'optimiser et d'améliorer les performances du déclencheur.

Corrigez immédiatement les exceptions et les erreurs du déclencheur

Les exceptions affectent gravement les performances. Si des exceptions ou des erreurs apparaissent dans le journal de débogage d'un déclencheur, corrigez immédiatement ces problèmes. S'il existe des exceptions qui ne peuvent pas être facilement évitées, vous pouvez essayer de les gérer avec un `try/catch` déclaration. Le `try/catch` l'instruction doit être encapsulée dans une petite fonction qui n'effectue aucune autre opération.

Dans l'exemple ci-dessous, `JSON.parse` la fonction entraîne une exception si l'entrée n'est pas dans une syntaxe JSON bien formée. Comme il n'existe aucun moyen de valider que la syntaxe JSON est bien formée avant l'analyse, cette tâche est encapsulée dans une petite fonction d'assistance .

```
function parseJSON(jsonString) {
    try {
        return JSON.parse(jsonString);
    } catch (e) {
        return null;
    }
}
let obj = parseJSON(HTTP.payload);
```

Conserver les actions là où elles sont nécessaires dans le script

Déplacez les actions, telles que les opérations sur les chaînes et l'accès aux propriétés, vers la branche la plus exclusive qui nécessite l' action.

```
// inefficient: The cookies variable may not be needed
let cookies = HTTP.cookies;
if (HTTP.method === 'POST') {
 // process cookies
}

// optimized: Cookies are not accessed unless they are needed
if (HTTP.method === 'POST') {
 let cookies = HTTP.cookies;
 // process cookies
}
```

Stockez localement les objets fréquemment consultés

Préservez les cycles du processeur en stockant localement les applications, les géolocalisations, les appareils et les autres objets auxquels on accède plusieurs fois.

```
// store object locally
let app = Application('example');
app.commit();
app.metricAddCount('custom', 1);
```

Évitez les variables dans l'instruction de débogage

N'incluez pas de variable dans l'instruction de débogage uniquement à des fins de débogage lorsque vous accédez à une charge utile ou à un autre objet volumineux. L'exclusion de variables de l'instruction de débogage vous permet de commenter la ligne de débogage et de conserver l'accès aux informations de charge utile.

N'incluez pas la fonction exit () dans le script

Insérez un `return` déclaration au lieu du `exit()` fonction globale, qui est obsolète en raison de mauvaises performances.

```
// avoid: slows the trigger down
HTTP.uri || exit();

// recommended
if (!HTTP.uri) return;
```

Note: Le comportement du `return` l'instruction est différente lorsqu'elle se trouve à l'intérieur d'une fonction imbriquée ; dans une fonction imbriquée, `return` L'instruction quitte uniquement la fonction en cours, pas l'intégralité du déclencheur.

N'incluez pas la fonction eval () dans le script

Les déclencheurs ne doivent pas tenter de générer du code de manière dynamique lors de l'exécution ; par conséquent, le `eval()` la fonction n'est pas prise en charge par l'API ExtraHop Trigger. L'inclusion de la fonction dans le déclencheur peut provoquer des erreurs d'exécution imprévisibles.

Prioriser les expressions régulières

Appliquez le `test()` méthode avec des expressions régulières au lieu de `match()` méthode, dans la mesure du possible. L'extraction d'une valeur à l'aide d'une expression régulière est souvent plus rapide que l'analyse complète du XML ou des chaînes de requête .

```
// slower search
let user = HTTP.parseQuery(payload).user;
```

```
// faster search
let match = /user=(.*?)\&/i.exec(payload);
let user = match ? match[1] : null;
```

Appliquer des opérateurs d'égalité stricts

Pour les comparaisons, appliquez des opérateurs d'égalité stricts, tels que triple égal (===) au lieu d'une égalité lâche (==), dans la mesure du possible. Les opérateurs d'égalité stricte sont plus rapides car ils ne tentent pas de coercition de type, par exemple en convertissant les adresses IP en chaînes avant de les comparer.

N'incluez pas de variables non déclarées

Déclarez des variables avec `let`, `var`, ou `const` instructions ; n'incluez jamais de variables non déclarées.

```
// inefficient: The variable "cookies" is never declared
cookies = HTTP.cookies;

// optimized: The variable "cookies" is declared
let cookies = HTTP.cookies;
```

Organiser le script avec des fonctions

Les fonctions sont bénéfiques pour les performances et l'organisation du code. Définissez les fonctions au début du déclencheur si celui-ci exécute la même opération à différents endroits dans le script du déclencheur ou si les opérations du déclencheur peuvent être séparées logiquement en parties distinctes.

Organiser les fonctions d'assistance en haut du script

Définissez les fonctions d'assistance en haut du script du déclencheur, plutôt qu'en bas, pour améliorer la lisibilité et aider le système ExtraHop à mieux comprendre et exécuter votre code.

Ajouter des tables de session

La table de session partage globalement des types de données simples (tels que des chaînes) entre tous les déclencheurs et flux. Respectez les meilleures pratiques suivantes lorsque vous ajoutez des tables de session à votre script de déclencheur.

Nombre d'incréments dans les tables de sessions

Lorsque vous comptez, appelez le `Session.increment` fonction au lieu de `Session.lookup` ou `Session.replace` fonctions. Le `Session.increment` la fonction est atomique et son exactitude est donc garantie sur plusieurs threads de déclenchement.

Ne collectez pas de mesures de comptage distinctes avec une table de session

Une table de session n'est pas un outil optimal pour compter le nombre de valeurs uniques, telles que les adresses IP, les ports ou les noms d'utilisateur, et elle est susceptible de provoquer des problèmes de performances. Au lieu de cela, validez une personnalisation décompte distinct métrique avec l'une des méthodes suivantes :

```
metricAddDistinct (
   name: String,
   item: String|Number|IPAddress
)

metricAddDetailDistinct (
   name: String,
    key: String|IPAddress,
   item: String|Number|IPAddress
)
```

Ne créez pas ou n'itérez pas de très grands tableaux ou objets

L'ajout de grands tableaux ou objets à une table de session peut réduire les performances. Par exemple, n'incluez pas plusieurs clés arrivant à expiration, qui ont une `notify=true` valeur, dans un déclencheur qui s'exécute sur `SESSION_EXPIRE` événement. De plus, n'appelez pas `JSON.parse` ou `JSON.stringify` méthodes sur des objets de grande taille dans le tableau de session.

Placer des valeurs dans le magasin Flow

Le magasin Flow collecte les valeurs des événements d'un seul flux, tel qu'une paire demande/réponse. Respectez les meilleures pratiques suivantes lorsque vous placez des valeurs dans le magasin Flow dans un script déclencheur.

Déterminer si les valeurs sont déjà disponibles

Avant de travailler avec le `Flow.store` propriété, vérifiez si les valeurs que vous souhaitez sont déjà disponibles directement depuis l'API ExtraHop Trigger. Les versions récentes du microprogramme ont supprimé le besoin d'un magasin Flow pour un certain nombre de propriétés de demande qui étaient couramment utilisées dans la réponse, telles que `HTTP.query` et `DB.statement`.

Ajoutez uniquement les valeurs nécessaires au magasin Flow

Évitez d'ajouter de gros objets au `Flow.store` propriété si vous n'avez besoin que d'un sous-ensemble des valeurs de l'objet. Par exemple, le code suivant ne stocke que `TTL` valeur de la propriété provenant du `DNS.answers` objet, au lieu de stocker toutes les valeurs de propriété du `DNS.answers` objet :

```
Flow.store.ttl = DNS.answers[0].ttl;
```

Effacer les valeurs une fois qu'elles ne sont plus nécessaires

Transparent `Flow.store` valeurs de propriété lorsqu'elles ne sont plus nécessaires en définissant la valeur de propriété sur `null`. L'effacement du magasin Flow permet d' éviter les erreurs dans lesquelles le déclencheur traite plusieurs fois la même valeur.

Ne partagez pas les valeurs du Flow Store entre les déclencheurs

N'accédez pas au `Flow.store` objet pour partager des valeurs entre différents déclencheurs exécutés sur le même événement. L'ordre dans lequel les déclencheurs s'exécutent lorsqu'un événement se produit n' est pas garanti. Si un déclencheur dépend de la valeur du `Flow.store` propriété d'un deuxième déclencheur, la valeur peut ne pas être celle attendue et peut donner des résultats incorrects dans le premier déclencheur.

Création d'enregistrements personnalisés

Les métriques personnalisées offrent la flexibilité nécessaire pour capturer les données dont vous avez besoin si les données ne sont pas déjà fournies par les métriques du protocole intégrées au système ExtraHop. Respectez les meilleures pratiques suivantes lors de la création de métriques personnalisées via un déclencheur.

Ne générez pas dynamiquement des noms de métriques personnalisés

La génération dynamique de noms de métriques personnalisés va dégrader les performances et peut empêcher la découverte automatique des métriques dans le catalogue de métriques.

Ne convertissez pas les adresses IP en chaînes avant de les ajouter à des métriques personnalisées

Si vous convertissez une adresse IP en chaîne avant de l'ajouter comme détail dans une métrique personnalisée, le système ExtraHop ne peut pas récupérer les attributs associés à l'adresse IP, tels que l'équipement ou le nom d' hôte associés.

```
// avoid
Application("SampleApp").metricAddDetailCount("reqs.byClient",
 Flow.client.ipaddr.toString(), 1);
```

```
// recommended
Application("SampleApp").metricAddDetailCount("reqs.byClient",
 Flow.client.ipaddr, 1);
```

Stockage des données dans des enregistrements

Les enregistrements vous permettent de stocker et de récupérer des informations structurées sur les transactions, les messages et les flux réseau. Respectez les meilleures pratiques suivantes lors de la création et du stockage d' enregistrements via un déclencheur.

Limiter l'accès aux propriétés via l'objet d'enregistrement

Évitez d'accéder aux propriétés via un objet d'enregistrement, tel que `HTTP.record`, si les propriétés sont disponibles dans l'objet de protocole ou dans l'objet Flow. Lorsque vous accédez à l'objet d'enregistrement, les données d'enregistrement sont allouées en mémoire. Accédez plutôt à l'objet d'enregistrement lorsque vous stockez des données dans un enregistrement personnalisé modifié.

```
// avoid
let uri = HTTP.record.uri;

// recommended
let uri = HTTP.uri;
```

Ne convertissez pas les adresses IP en chaînes avant de les stocker dans des enregistrements

Le système ExtraHop filtre et trie par adresses IP, qui ne peuvent pas être appliquées à des chaînes telles que les blocs CIDR pour les adresses IPv4.

Accès à la banque de données

Les déclencheurs Datastore vous permettent d'accéder à des mesures agrégées qui ne sont pas disponibles pour les événements basés sur les transactions, tels que `HTTP_REQUEST`. Respectez les bonnes pratiques suivantes lorsque vous accédez à la banque de données via un déclencheur.

Créez des déclencheurs de banque de données avec parcimonie

Les ressources des déclencheurs de la banque de données sont très limitées. Dans la mesure du possible, évitez d'exécuter des déclencheurs sur les événements suivants de la banque de données :

- ALERT_RECORD_COMMIT
- METRIC_RECORD_COMMIT
- DÉBUT DU CYCLE_MÉTRIQUE
- FIN DE CYCLE MÉTRIQUE

Par exemple, si vous souhaitez uniquement accéder périodiquement à une table de session, exécutez le déclencheur sur le `SESSION_EXPIRE` événement au lieu du `METRIC_CYCLE_END` événement.

Configuration des options avancées lors de la validation d'enregistrements métriques

Assurez-vous que options de déclencheur avancées sont configurés pour les déclencheurs qui s'exécutent sur `METRIC_RECORD_COMMIT` événement. Plus précisément, nous vous recommandons de configurer un cycle métrique de 30 secondes pour les déclencheurs qui s'exécutent sur `METRIC_RECORD_COMMIT` événement.

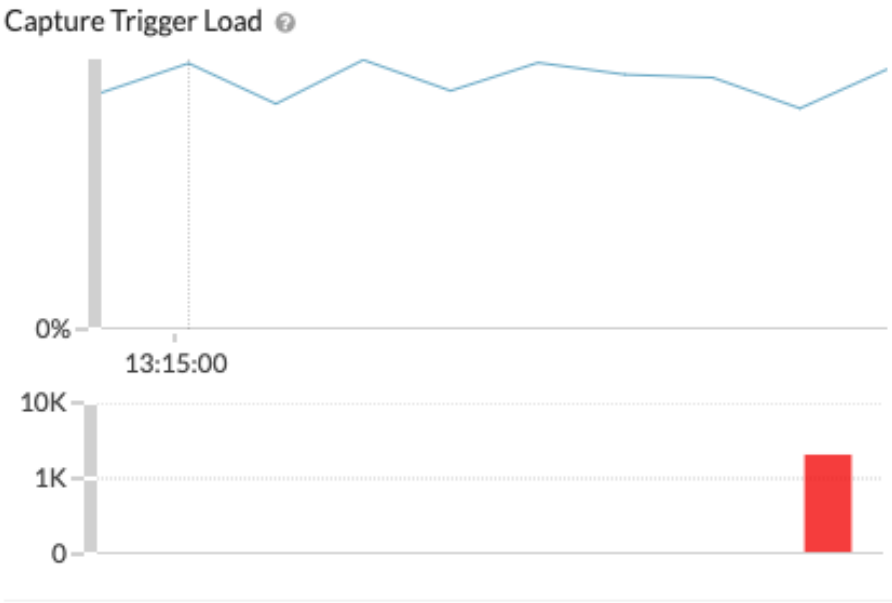
Évaluez les performances du déclencheur

Vous pouvez surveiller et évaluer l'impact des déclencheurs en cours d'exécution sur votre système ExtraHop de plusieurs manières. Vous pouvez consulter les informations de performance d'un déclencheur individuel et vous pouvez consulter plusieurs graphiques qui indiquent l'impact collectif de tous vos déclencheurs sur le système.

Afficher les performances d'un déclencheur individuel

Vous pouvez vérifier si un déclencheur contient des erreurs ou des exceptions dans l'onglet Journal de débogage du volet Modifier le déclencheur. Accédez à la page Déclencheurs en cliquant sur l'icône des paramètres système dans le système ExtraHop.

Vous pouvez consulter le coût de performance d'un déclencheur en cours d'exécution sur le graphique Capture Trigger Load dans l'onglet Modifier le déclencheur. Le graphique affiche un graphique des performances du déclencheur qui suit le nombre de cycles consommés par le déclencheur au cours d'un intervalle de temps donné. Vous pouvez survoler un point de données pour afficher les indicateurs de performance clés à un moment précis.



La pointe du pointeur inclut les informations suivantes :

- Le plus et le moins de cycles sont utilisés par le déclencheur pour traiter un seul événement.
- Le nombre de fois que le déclencheur s'est exécuté et le pourcentage de fois où il s'est exécuté par rapport à tous les déclencheurs exécutés au cours de la même période.
- Le nombre total de cycles consommés par le déclencheur et le pourcentage de cycles consommés par rapport à tous les déclencheurs exécutés dans le même intervalle de temps.

Afficher les performances de tous les déclencheurs du système

La page System Health contient plusieurs graphiques qui fournissent une vue d'ensemble des déclencheurs exécutés sur le système ExtraHop. Accédez à la page System Health en cliquant sur l'icône des paramètres système dans le système ExtraHop.

En consultant les graphiques d'état du système décrits dans les directives suivantes, vous pouvez surveiller les problèmes susceptibles d'affecter les performances du système ou d'entraîner des données incorrectes.

Vérifiez que le déclencheur est en cours d'exécution

Le Détails du déclencheur le graphique affiche tous les déclencheurs en cours d'exécution sur le système. Si le déclencheur que vous venez de créer ou de modifier ne figure pas dans la liste, il se peut que vous rencontriez un problème avec le script du déclencheur.

Surveillez les activités inattendues

Le Le déclencheur s'exécute et s'arrête un graphique peut afficher des rafales d'activité des déclencheurs susceptibles d'indiquer un comportement inefficace dû à un ou plusieurs déclencheurs. Si des rafales sont affichées, consultez le Le déclencheur s'exécute par déclencheur graphique permettant de localiser tout déclencheur consommant plus de ressources que la moyenne, ce qui peut indiquer qu'un script mal optimisé du déclencheur affecte les performances.

Vérifiez les exceptions de déclencheur non gérées

Le Déclenchez des exceptions par déclencheur le graphique affiche toutes les exceptions causées par des déclencheurs. Les exceptions contribuent dans une large mesure aux problèmes de performance du système et doivent être corrigées immédiatement.

Vérifiez les déclencheurs supprimés de la file d'attente

Le Le déclencheur s'exécute et s'arrête le graphique affiche le nombre de déclencheurs qui ont été supprimés de la file d'attente des déclencheurs. L'une des causes fréquentes d' abandon des déclencheurs est un déclencheur de longue durée qui domine la consommation de ressources. Un système en bonne santé ne doit contenir aucune goutte à tout moment.

Surveiller la consommation des ressources

Le Charge du déclencheur Le graphique suit l'utilisation de toutes les ressources disponibles par déclencheurs. Une charge élevée est d'environ 50 %. Recherchez les pics de consommation qui peuvent indiquer qu'un nouveau déclencheur a été introduit ou qu'un déclencheur existant présente des problèmes.

Vous pouvez vérifier si les déclencheurs de votre banque de données, également appelés déclencheurs de pont, fonctionnent correctement à l'aide des graphiques suivants :

- Le déclencheur de la banque de données s'exécute et s'arrête
- Chargement déclencheur de la banque de données
- Exceptions de déclenchement de la banque de données par déclencheur

Voir le FAQ sur l'état du système pour passer en revue les questions fréquemment posées sur la manière dont les tableaux de santé du système peuvent vous aider à évaluer les performances du déclencheur sur votre système ExtraHop.

Afficher les ressources du déclencheur

Pour en savoir plus sur les déclencheurs et l'API ExtraHop Trigger, consultez les ressources suivantes :

Documentation

- Référence de l'API ExtraHop Trigger
- Commencez avec les déclencheurs section du Guide de l'interface utilisateur Web.

Procédures pas à pas

- Procédure pas à pas : créez un déclencheur pour collecter des métriques personnalisées pour les erreurs HTTP 404
- Procédure pas à pas de l'UPA : création d'un déclencheur pour surveiller les réponses aux demandes NTP monlist

- Présentation d'un flux de données ouvert : configurez un flux de données ouvert pour envoyer des données métriques à AWS Cloudwatch

Entraînement

- Planification d'un déclencheur
- Création d'un déclencheur simple
- Création et utilisation de conteneurs d'applications
- Création d'un déclencheur multi-événements

Communauté

- Les forums de la communauté ExtraHop