# Erstellen Sie eine benutzerdefinierte Erkennung

Veröffentlicht: 2025-03-28

Mit benutzerdefinierten Erkennungen können Sie Kriterien angeben, anhand derer Erkennungen auf dem ExtraHop-System generiert werden. Maschinelles Lernen und regelbasierte Erkennungen erfassen ungewöhnliche Verhaltensweisen und häufige Bedrohungen. Durch die Erstellung einer benutzerdefinierten Erkennung können Sie jedoch die Geräte und Verhaltensweisen genauer untersuchen, die für Ihr Netzwerk von entscheidender Bedeutung sind.

Wenn Sie eine benutzerdefinierte Erkennung erstellen, müssen Sie einen Auslöser erstellen, der das Systemereignis und die Bedingungen identifiziert, auf die das System achten soll, und dann können Sie den Auslöser den spezifischen Geräten oder Gerätegruppen zuweisen, die Sie überwachen möchten. Wenn das Ereignis eintritt, wird eine Erkennung generiert.

In diesem Handbuch finden Sie die Schritte und ein Beispielskript, das eine benutzerdefinierte Erkennung generiert, wenn verdächtige Verbindungen zu bestimmten Websites über Windows PowerShell hergestellt werden.

**Bevor Sie beginnen**

- Sie müssen mit ExtraHop vertraut sein Trigger. Betrachten Sie insbesondere diese Best Practices beim Schreiben Ihres Skripts und beim Zuweisen von Triggern.
- Sie benötigen ein Benutzerkonto bei Privilegien erforderlich, um Trigger zu erstellen.
- Wenn du eine hast Konsole, erstelle einen Auslöser auf dem Konsole und der Auslöser läuft auf allen angeschlossenen Sensoren.

## Einen Auslöser erstellen, um benutzerdefinierte Erkennungen zu generieren

Trigger generieren benutzerdefinierte Erkennungen, indem sie den aufrufen `commitDetection` Funktion im Trigger-Skript.

Im folgenden Beispiel generiert der Auslöser eine benutzerdefinierte Erkennung, wenn ein PowerShell-Client eine Website aufruft, die als Staging-Site für exfiltrierte Daten bekannt ist.

Der Auslöser identifiziert PowerShell-Verbindungen, indem er nach JA3-Hashes des TLS-Clients sucht, die bekannten PowerShell-Clients gehören.

Wenn die TLS-Verbindung von einem PowerShell-Client zu einem verdächtigen Host hergestellt wird, generiert der Auslöser eine Erkennung. Die Erkennung umfasst die Version von PowerShell, die die Verbindung initiiert hat, die Server-IP-Adresse und die Client-IP-Adresse.

**Hinweis:** Für weitere Informationen über die `commitDetection` Funktion, siehe Trigger-API-Referenz.

1. Klicken Sie auf das Symbol Systemeinstellungen und klicken Sie dann **Trigger**.
2. klicken **Erstellen**.
3. Geben Sie die folgenden Trigger-Konfigurationseinstellungen an:

   **Name**

   Geben Sie einen Namen für Ihren Auslöser ein. Dieser Name identifiziert Ihren Auslöser, nicht die Erkennung.

   In unserem Beispiel geben wir den Namen ein: `Benutzerdefinierte Erkennung: PowerShell-Verbindung zur verdächtigen Site`.

**Beschreibung**

(Optional) Geben Sie die Beschreibung des Auslöser ein. Diese Beschreibung bezieht sich auf den Auslöser, nicht auf die Erkennung.

In unserem Beispiel geben wir die Beschreibung ein: `Erzeugt jedes Mal eine Erkennung, wenn ein PowerShell-Client eine Verbindung zu pastebin, raw.githubusercontent.com oder Githack herstellt. PowerShell-Clients werden durch JA3-Hashes identifiziert.`

**Ereignisse**

Wählen Sie das Ereignis aus, bei dem der Auslöser ausgeführt wird.

In unserem Beispiel wählen wir das Ereignis SSL_OPEN aus. Dieses Ereignis tritt ein, wenn zum ersten Mal eine TLS-Verbindung hergestellt wird.

**Zuweisungen**

Wählen Sie das Gerät oder die Gerätegruppe aus, die Sie überwachen möchten. Weisen Sie Ihren Auslöser zunächst einem einzelnen Gerät zum Testen zu. Nachdem Sie bestätigt haben, dass die benutzerdefinierte Erkennung ordnungsgemäß funktioniert, weisen Sie den Auslöser einer Gerätegruppe zu, die alle Geräte enthält, die Sie überwachen möchten.

Da PowerShell ein Windows-Befehlszeilentool ist, wählen Sie einen Microsoft-Server aus, um den Auslöser zu testen. Nachdem Sie bestätigt haben, dass die benutzerdefinierte Erkennung ordnungsgemäß funktioniert, ändern Sie die Zuweisung zu einer Gerätegruppe, die alle Ihre wichtigen Microsoft-Server enthält. Weitere Informationen zum Erstellen von Gerätegruppen finden Sie unter Erstellen Sie eine Gerätegruppe.

4. Geben Sie im rechten Bereich den Code ein, der bestimmt, wann Ihre benutzerdefinierte Erkennung generiert wird.

   In unserem Beispiel identifiziert der folgende Triggercode, wenn ein Client eine Verbindung zu pastebin, githubusercontent oder githack initiiert:

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

}
```

5. Geben Sie als Nächstes den Code ein, der Ihre benutzerdefinierte Erkennung festlegt. Das `commitDetection` Die Funktion muss im folgenden Format geschrieben werden:

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

Geben Sie Werte für jeden der folgenden Parameter in Ihrem Skript ein.

| Wert | Beschreibung |
|---|---|
| Erkennungstyp-ID | Eine eindeutige Zeichenfolge, die Ihre benutzerdefinierte Erkennung identifiziert. Diese Zeichenfolge darf nur Buchstaben, Zahlen und Unterstriche enthalten. |

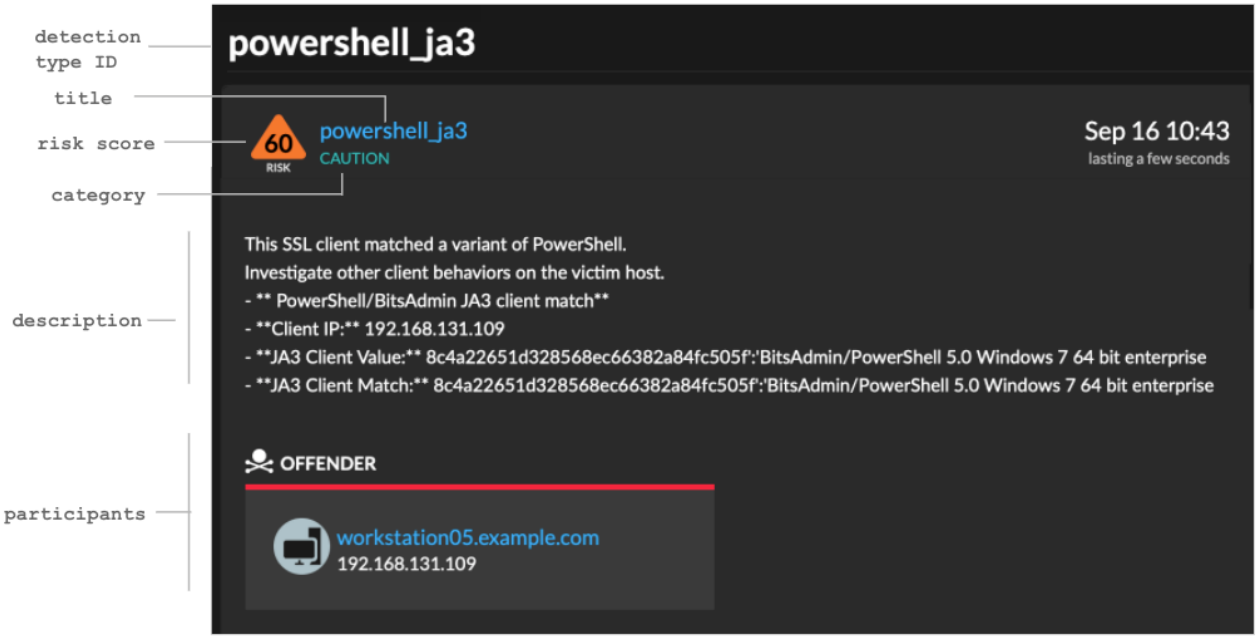
| Wert | Beschreibung |
| --- | --- |
| Titel | Text, der oben auf der Erkennungskarte erscheint. Geben Sie einen aussagekräftigen Titel ein, der leicht zu scannen ist.<br><br>Dieser Titel erscheint im Erkennungskatalog als Anzeigename für Ihren Erkennungstyp, gefolgt von `[benutzerdefiniert]`. |
| Beschreibung der Erkennung | Text, der unter dem Titel und der Kategorie auf einer Erkennungskarte erscheint. Geben Sie Informationen über das Ereignis ein, das die Erkennung generiert.<br><br>Dieses Feld unterstützt Markdown. Wir empfehlen, Interpolationsvariablen einzubeziehen, um spezifische Informationen zu Ihrer Erkennung anzuzeigen.<br><br>Zum Beispiel die Variablen `$(Flow.client.ipaddr)` und `$(Flow.server.ipaddr)` die IP-Adresse des Client- und Servergeräts im Fluss anzeigen und `$(Flow.l7proto)` zeigt das L7-Protokoll an. Einschließen `\n` am Ende jeder Textzeile, um sicherzustellen, dass die Beschreibung korrekt angezeigt wird. |
| Risikoscore | Eine Zahl, die die Wahrscheinlichkeit, Komplexität und geschäftliche Auswirkungen einer Sicherheitserkennung misst. Das Symbol für die Risikobewertung wird oben auf der Erkennungskarte angezeigt und ist nach Schweregrad als rot (80-99), orange (31-79) oder gelb (1-30) farblich gekennzeichnet. Du kannst Entdeckungen nach Risiko sortieren. |
| Beteiligter des Täters<br><br>Teilnehmer des Opfers | Eine Reihe von Objekten, die die Teilnehmer an der Erkennung identifiziert. Beispielsweise identifiziert das folgende Array in einem Fluss den Server als Täter und den Client als Opfer:<br><br>`participants: [Flow.client.victim, Flow.server.offender]`<br><br>Weitere Informationen zu Gerät, IP-Adresse und Anwendungsobjekten finden Sie in der Trigger-API-Referenz. |
| Identitätsschlüssel | Eine Zeichenfolge, die die Identifizierung laufender Erkennungen ermöglicht. Wenn mehrere Erkennungen mit demselben Identitätsschlüssel und Erkennungstyp innerhalb des von der angegebenen Zeitspanne generiert werden `identityTtl` Parameter, die Erkennungen werden zu einer einzigen fortlaufenden Erkennung zusammengefasst. |

| Wert | Beschreibung |
| --- | --- |
| | Erstellen Sie eine eindeutige Identitätsschlüsselfolge, indem Sie die Merkmale der Erkennung kombinieren.<br><br>Beispielsweise wird der folgende Identitätsschlüssel erstellt, indem die Server-IP-Adresse und die Client-IP-Adresse kombiniert werden:<br><br>`identityKey: [Flow.server.ipaddr, Flow.client.ipaddr].join('!!')` |
| Zeitraum | Der Zeitraum nach dem Generieren einer Erkennung, in dem doppelte Erkennungen zu einer fortlaufenden Erkennung zusammengefasst werden. Der Zeitraum wird zurückgesetzt und die Erkennung endet erst, wenn der Zeitraum abgelaufen ist.<br><br>Die folgenden Zeiträume sind gültig:<br><br>• `hour`<br>• `day`<br>• `week`<br><br>Der Standardzeitraum ist `hour`. |

Das folgende Beispiel zeigt den abgeschlossenen Skriptabschnitt.

```
commitDetection('powershell_ja3', {
        title:
'PowerShell / BitsAdmin Suspicious Connection',
        description:
"This TLS 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',
});
```

Diese Werte werden auf der Erkennungskarte ähnlich der folgenden Abbildung angezeigt:




6. klicken **Speichern** und klicken Sie dann **Erledigt**.

   siehe Beispiel für einen benutzerdefinierten Erkennungsauslöser für ein vollständiges kommentiertes Skript.

Ihre benutzerdefinierte Erkennung wird dem Erkennungskatalog hinzugefügt, nachdem Ihr Auslöser zum ersten Mal ausgeführt wird. Erkennungskategorien und MITRE-Techniken hinzufügen zur Erkennung aus dem Erkennungskatalog.

## Erstellen Sie einen benutzerdefinierten Erkennungstyp

Nachdem Sie einen Auslöser zum Generieren Ihrer benutzerdefinierten Erkennung erstellt haben, können Sie im Erkennungskatalog einen benutzerdefinierten Erkennungstyp erstellen, um weitere Informationen zu Ihrer Erkennung hinzuzufügen.

Sie können einen Anzeigenamen angeben und Erkennungskategorien hinzufügen, damit Sie Ihre Erkennung auf der Seite „Entdeckungen“ leichter finden können. Sie können auch MITRE-Links hinzufügen, sodass Ihre benutzerdefinierte Erkennung in der Matrix auf der Seite Nach MITRE Technique gruppieren angezeigt wird.

1. Loggen Sie sich in das ExtraHop-System ein über `https://<extrahop-hostname-or-IP-address>`.
2. Klicken Sie auf das Symbol Systemeinstellungen und klicken Sie dann **Erkennungskatalog**.
3. Führen Sie auf der Seite Erkennungskatalog einen der folgenden Schritte aus:
   - Wenn Ihr Auslöser bereits ausgeführt wurde, fügt das System Ihre benutzerdefinierte Erkennung automatisch zum Katalog hinzu, wobei der im Auslöser angegebene Anzeigename vorangestellt ist `[benutzerdefiniert]`. Klicken Sie auf den Erkennungstyp, um ihn zu bearbeiten.
   - Wenn Ihr Erkennungstyp nicht erstellt wurde, klicken Sie auf **Erstellen**.
4. Füllen Sie die folgenden Felder aus:

   **Name anzeigen**

   Geben Sie einen eindeutigen Namen für den Titel der Erkennung ein.

**Erkennungstyp-ID**

Geben Sie den Wert ein, den Sie für die Erkennungstyp-ID im Auslöser eingegeben haben. Wenn Sie beispielsweise Folgendes eingegeben haben: `commitDetection('network_segmentation_breach')`, die Erkennungstyp-ID lautet „network_segmentation_breach“. Sie können die Erkennungstyp-ID nicht bearbeiten, nachdem der Erkennungstyp gespeichert wurde.

**Autor**

Geben Sie den Autor der benutzerdefinierten Erkennung ein.

**MITRE-Technik**

Wählen Sie im Dropdownmenü eine oder mehrere MITRE-Techniken aus, die Sie mit der Erkennung verknüpfen möchten.

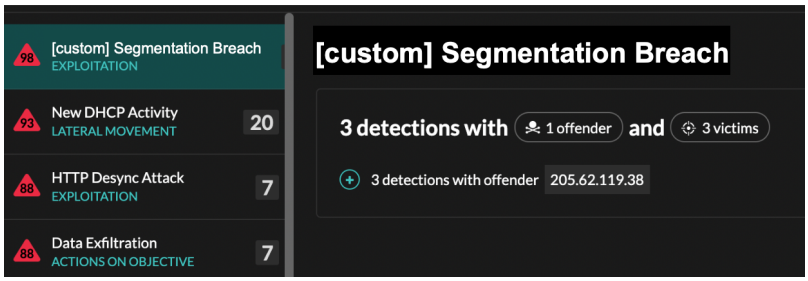
5. klicken **Speichern**.

## Benutzerdefinierte Erkennungen anzeigen

Sie können benutzerdefinierte Erkennungen auf der Erkennungen Seite mit anderen integrierten Erkennungen.

Gruppieren Sie die Erkennungsseite nach Typ. Alle Entdeckungen in der Erkennungsliste sind nach Erkennungstyp gruppiert.

Zum Beispiel, wenn Ihr Erkennungsanzeigename lautet `[custom]Segmentation Breach`, würde der Eintrag in der Erkennungsliste ähnlich der folgenden Abbildung erscheinen:




Wählen Sie links oben auf der Seite **MITRE Karte**. Die MITRE-Techniken, die mit der benutzerdefinierten Erkennung verknüpft wurden, sind in der Matrix hervorgehoben.

### Die nächsten Schritte

Eine Regel für Erkennungsbenachrichtigungen erstellen. Sie können das ExtraHop-System beispielsweise so konfigurieren, dass es Ihnen eine E-Mail sendet, wenn Ihre benutzerdefinierte Erkennung erfolgt.

## Beispiel für einen benutzerdefinierten Erkennungsauslöser

Das folgende Skript ist das vollständige PowerShell/JA3-Beispiel, auf das in diesen Anweisungen verwiesen wird.

```
// If the server is internal, exit
if ( ! Flow.server.ipaddr.isExternal ) {
    return;
}
// If the TLS host name is not set, exit
if(SSL.host === null) { return; }
```

```
// Continue only if the TLS hostname belongs to one of the suspicious sites
if(SSL.host.match(/pastebin/i) || SSL.host.match(/raw.githubusercontent.com/
i) || SSL.host.match(/githack/i)) {

    // List of common PowerShell JA3 hashes
    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',
    }));
    // Store the client JA3 hash in a variable
    const hash = SSL.ja3Hash;

    // Iterate through each PowerShell JA3 hash
    for ( let ja3 in suspect_ja3_hashes ) {

        // If the client JA3 hash is from PowerShell,
        // commit the detection
        if ( hash.includes(ja3) ) {

            commitDetection('PowerShell_JA3', {
                categories: ['sec.caution'],
                title: "PowerShell / BitsAdmin Suspicious Connection",
                // Specify the offender as the device object of the client
                participants: [Flow.client.offender],
                description:
                    "This TLS 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"+
                    "- **Server IP:** " + Flow.server.ipaddr + "\n"+
                    "- **JA3 Client Value:** " + ja3 + "\n"+
                    "- **JA3 Client Match:** " + suspect_ja3_hashes[ja3],
                // Create the identity key by combining the server IP
 address, client IP address, and PowerShell JA3 hash
```

```
                identityKey: [
                    Flow.server.ipaddr,
                    Flow.client.ipaddr,
                    hash
                ].join('!!'),
                riskScore: 60,
                identityTtl: 'hour'
            });
        }
    }
}
```